SEMANTIC TECHNOLOGY FOR
INTELLIGENCE, DEFENSE, AND SECURITY

STIDS 2013

Semantic Technologies for Big Data

THE 8TH INTERNATIONAL CONFERENCE
ON SEMANTIC TECHNOLOGIES
NOVEMBER 12-15, 2013

Mason Inn Conference Center
George Mason University
Fairfax, Virginia Campus

Conference Proceedings

Kathryn B. Laskey
Ian Emmons
Paulo C. G. Costa
(Eds.)

GEORGE MASON UNIVERSITY

C4I CENTER

DATA TACTICS CORPORATION

Preface

The 8th International Conference on Semantic Technologies for Intelligence, Defense, and Security (STIDS 2013) provides a forum for academia, government and industry to share the latest research on semantic technology for defense, intelligence and security applications.

Semantic technology is a fundamental enabler to achieve greater flexibility, precision, timeliness and automation of analysis and response to rapidly evolving threats.

The STIDS 2013 theme is Semantic Technologies for Big Data.

Topics of general interest for STIDS include:

- Creating an interoperable suite of public-domain ontologies relevant to intelligence analysis covering diverse areas
- Ontologies and reasoning under conditions of uncertainty
- Semantic technology and ontological issues related to:
    - Source credibility and evidential pedigree
    - Use of sensing devices including security, e.g. global infrastructure grid (GIG), images and intelligence collection in general
- Usability issues relating to semantic technology
- Best practices in ontological engineering

Fairfax, VA
November 2013

Ian Emmons and Kathryn Laskey
STIDS 2013 Technical Chairs

Paulo Costa
STIDS 2013 General Chair

STIDS 2013 Committees

STIDS 2013 Program Committee

Stephen Allen	Semantic Research, Inc.
Carl Andersen	Raytheon BBN Technologies
Robert Battle	Amazon.com
Rommel Carvalho	George Mason University
Werner Ceusters	University at Buffalo
Suzanne Collier	Raytheon BBN Technologies
Paulo Costa	George Mason University
Mike Dean	Raytheon BBN Technologies
Jody Des Roches	Joint Warfare Analysis Center
Ian Emmons	Raytheon BBN Technologies
Matt Fisher	Progeny Systems Corporation
Katherine Goodier	Xcelerate Solutions
Mark Greaves	Pacific Northwest National Lab
Richard Haberlin	George Mason University
Brian Haugh	Institute for Defense Analyses
John Hebeler	University of Maryland - BC
Terry Janssen	SAIC, Inc.
Greg Joiner	Raytheon BBN Technologies
Kenneth Kisiel	Office of Naval Research
Mieczylaw Kokar	Northeastern University
Dave Kolas	Raytheon BBN Technologies
Kathryn Laskey	George Mason University
Nancy Lawler	US Department of Defense
Mike Letsky	Office of Naval Research
William Mandrick	Data Tactics, Inc.
Dan Maxwell	KaDSci, Inc.
Dave Mireles	Raytheon BBN Technologies
Ranjeev Mittu	US Navy Research Laboratory
Jeffrey Morrison	Office of Naval Research
Leo Obrst	MITRE Corporation
Mary Parmelee	MITRE Corporation

Andrew Perez-Lopez	Opower, Inc.
Plamen Petrov	Raytheon BBN Technologies
Setareh Rafatirad	George Mason University
Doug Reid	Google, Inc.
Joe Rockmore	Cyladian Technology Consulting
Dorene Ryder	Raytheon BBN Technologies
Ciara Sibley	US Navy Research Laboratory
Barry Smith	NCOR, University at Buffalo
Tony Stein	Raytheon BBN Technologies
Gheorghe Tecuci	George Mason University
Andreas Tolk	Old Dominion University
Brian Ulicny	Vistology, Inc.
Andrea Westerinen	SAIC
Duminda Wijesekera	George Mason University

STIDS Steering Committee

Paulo Costa	George Mason University
Mike Dean	Raytheon BBN Technologies
Ian Emmons	Raytheon BBN Technologies
Katherine Goodier	NIC, Inc.
Terry Janssen	SAIC
Kathryn Laskey	George Mason University
William Mandrick	Data Tactics
Leo Obrst	MITRE Corporation
Barry Smith	NCOR, University at Buffalo

STIDS 2013 Organizing Committee

General Chair

Paulo Costa

Technical Chairs

Ian Emmons

Kathryn Laskey

Publicity Chair

William Mandrick

Classified Session Chair

Brian Haugh

Local Team (GMU)

Debra Schenaker (Administrative Chair)

Priscilla McAndrews

Shou Matsumoto

Felipe Bombarda

Karen Tai

STIDS 2012 Platinum Sponsor

Data Tactics Corporation is a minority-owned small business that specializes in Data Management, Data Architecture, Data Engineering, Semantic Data Representations, and Big Data. Since 2005, our engineers have been on the forefront of large, multi-domain, data systems supporting Government and commercial organizations. Our engineering staff is over 90% TS/SCI cleared (many with polygraphs) with over 25% having advanced degrees and doctorates.

We offer a suite of solutions to help customers handling very large, "Big Data" problem sets. Our team of senior engineers and data scientists excel at the most intractable problems for customers such as AIR FORCE, ARMY, DARPA, DHS, DNI, NSA and many others. From tactical to strategic efforts, our team has led the creation, integration, and implementation of innovative and proven solutions in the world of Data Alignment, Modeling, and Analytics. We are also very active in standards development including the NIST Cloud Computing and Big Data standards along with Semantic Standards (e.g. BFO, SUMO, DOLCE, etc) and actively contribute to the open source communities (e.g. Apache, Source Forge, GIT, etc).

Data Tactics is highly invested in fostering and/or leading collaborations with academia and national labs in advanced research & development initiatives that support disruptive technologies. Our team brings a rich history of supporting prototyping, experimental technology integration, mission oriented demonstrations, and specifically cloud development and integration.

DATA TACTICS – WHAT WE DO

CLOUD/DISTRIBUTED COMPUTING REFERENCE ARCHITECTURES

- IC ITE DNI Enterprise Strategy
- Army Red Disk/NSA Ghost Machine
- DCGS-A Standard Cloud (DSC)
- Air Force TENCAP

TURNKEY BIG DATA IMPLEMENTATIONS

- Secure Enterprise Hadoop
- Elastic Ingest and Semantic Markup
- Distributed Analytics

ADVANCED ANALYTICS

- Multiple Algorithm Development Programs
- Information Artifact Ontology Development
- Advanced Machine Learning (i.e. NLP) integration
- Advanced Video and Image Entity extraction

SECURE DATABASE ARCHITECTURES

- Secure Entity Database (SED)
- Defense Cross-Domain Analytic Capability (DCAC)

CYBERSECURITY

- Information Assurance
- Security Architecture, Design, and Configuration
- Policies, Process Development, and Validation

STIDS 2012 Keynote Speaker: November 13

Dr. Benjamin Grosof

Highly Expressive yet Scalable Knowledge for Intelligence, Defense, and Security

We present recent results on semantic web knowledge representation & reasoning, and knowledge acquisition, that tightly combine highly expressive rules and ontologies specified semi-automatically -- yet rapidly -- by starting from effectively unrestricted English text. The knowledge employs a new logic, Rulelog, that has strong capabilities to express meta knowledge. Rulelog permits higher-order logic formulas that are defeasible (i.e., can have exceptions). It is rich enough to serve as a relatively direct target for natural language processing, using Textual Logic, a new method that employs logic-based mappings in natural language (NL) text interpretation and text generation.

Rulelog also leverages its meta capabilities to achieve computational tractability via restraint, a new form of bounded rationality. Rulelog, and the Textual Logic that leverages it, constitute a pretty radical step forward in terms of fundamental capabilities in semantic tech, with a number of advantages including in the social scalability of knowledge authoring and reuse. Yet this step is incremental relative to legacy technology, in that the new knowledge representation transforms into the same fundamental logic as used in relational and RDF databases and in commercially predominant business rule systems. It's applicable in several intelligence, defense, and security (IDS) areas including: info access policies (e.g., confidentiality, compliance); info integration, flow and ontology mapping (e.g., in situation awareness); modeling of causal events and risk; intelligence analysis and debate; e-learning (e.g., just-in-time training); contracts (e.g., compliance); question-answering (QA); and NL-based human-computer interaction (HCI).

We discuss some of the exciting opportunities and challenges.

Biography: Dr. Benjamin Grosof

Benjamin Grosof is an industry leader in knowledge representation, reasoning, and acquisition. He has pioneered semantic technology and industry standards for rules, the combination of rules with ontologies, the applications of rules in e-commerce and policies, and the acquisition of rules and ontologies from natural language (NL). He has had driving roles in RuleML, W3C RIF (Rule Interchange Format), and W3C OWL-RL (rule-based ontologies). He led the invention of several fundamental technical advances in knowledge representation, including courteous defeasibility, restraint bounded rationality, and the rule-based technique, which rapidly became the currently dominant approach to commercial implementation of OWL. He has extensive experience in machine learning, probabilistic reasoning, and user interaction design.

Dr. Grosof has experience applying core technology for knowledge, reasoning, and related HCI in a wide variety of application areas, including: trust/privacy/security, contracts, compliance, legal, and services engineering; financial/ insurance services, risk management,and regulations; defense and national intelligence; biomedical research; and data/ decision analytics. From fall 2007 to early 2013, he led a large research program in Artificial Intelligence (AI) and rule-based semantic technologies at Vulcan Inc. for Paul G. Allen; this centered around the SILK system for highly expressive, yet scalable, rules. Previously he was an IT professor at MIT Sloan (2000-2007) and a senior software scientist at IBM Research (1988-2000). He is president of the expert consulting firm Benjamin Grosof & Associates founded while he was at MIT, and co-founder of the recent start-up Coherent Knowledge Systems.

His background includes 4 major industry software releases, 2 years in software startups, a Stanford PhD (Computer Science), a Harvard BA (Applied Mathematics), 2 patents, and over 50 refereed publications.

STIDS 2013 Keynote Speaker: November 14

Dr. Jeffrey Morrison

Exploring the role of Context in Applied Decision Making

Decision makers in operational environments are often surprised by emerging events and have little time to give deep consideration to alternative courses of action before being forced to make a decision. Decision support has evolved over the last 20 years but even today, decision support tools do not dynamically adapt to a decision maker's context. This often results in less than optimal decision making. Recent advances in the fields of cognitive science, the mathematics of decision science, human behavioral modeling, team decision making, knowledge creation and transfer, mental model processes, semantic techniques and human factors present new opportunities to create decision support that is context sensitive, and potentially, proactive. To accomplish this, a systematic exploration of the role of context needs to be studied in decision support systems that enable operational decision making.

Decision making is challenging for a number of reasons. Finding and integrating decision-relevant information is hard. Context is often absent, implicit, sparsely or poorly represented in task environments requiring its laborious and error-prone internal reconstruction by decision makers. The modern pace of operations often means that warfighters find themselves engaging in tasks in ways, and in combinations, for which they hadn't planned, and for which they may not be prepared. This forces decision makers to multi-task amongst many competing and often conflicting mission objectives concurrently.

Next generation decision support will not just "get the decision maker in the ball park" but will be proactive in trying to "keep the decision maker in the ball park"

throughout the process despite the high levels of uncertainty and highly dynamic environments. At the center of this new research initiative is the idea that we can develop technologies that are contextually aware of a decision makers' missions and tasks. It is asserted that algorithms can be developed that effectively anticipate the decision and information needs of decision makers, in many kinds of task environments. Algorithms would then enable the timely presentation of information. Enabling machines to dynamically model and share context with the human decision makers will be key to enabling Proactive Decision Support (PDS). Such decision support will enable the recognition of changes in the environment and the implications for shifting priorities for decisions that could address operational complexity and make enable decision makers to make more optimal decisions, faster.

Biography: Dr. Jeffrey Morrison

Dr. Jeffrey G. Morrison joined ONR's Human & Bioengineered Systems Department (341) as a Program Officer in January 2011 where he leads the Command Decision Making (CDM) program. The program is conducting Basic & Applied cognitive science research for application to individual & group decision making. The current operational focus is on multi-echelon Command & Control. The science focus is on developing Proactive Decision Support tools (PDS) that are aware of mission and tasks context as well as the facilitating the development of a science of Context-Driven Decision Making (CDDM).

Prior to coming to ONR, Dr. Morrison was an Engineering Psychologist / Cognitive Scientist with the Space and Naval Warfare Systems Center – Pacific (SSC Pacific) for 17 years. He was most recently embedded as a Navy Scientist with the Combating Terrorism Technical Support Office (CTTSO) where he served as Chief Scientist to the ASD RDT&E sponsored Human Social Culture and Behavior Modeling Program (HSCB). During 2007-2008, Dr. Morrison was detailed to the Director of National Intelligence where he served as an IARPA Program Manager studying the analytic process and the potential application of virtual world technologies to enable it. Dr. Morrison was a senior scientist supporting several DARPA projects, including the development of user-composable automation for Maritime Domain Awareness (FastC2AP), Predictive Analysis for Naval Deployment Activity (PANDA), and the Augmented Cognition program. He also was principle investigator for numerous ONR sponsored projects, including: Knowledge Web (K-Web), and Tactical Decision Making Under Stress (TADMUS).

Dr Morrison has been the recipient of numerous professional awards including: The 2005 Jerome H Ely Award for Article of the Year in the Journal of Human Factors; the 2004 ONR Arthur E. Bisson Prize for Naval Technology Achievement; and the American Psychological Association - Division 21, George E. Briggs Award for Original Research.

Table of Contents

Preface ..... *i-ix*

Technical Papers

Context Correlation Using Probabilistic Semantics
Setareh Rafatirad, Kathryn Laskey and Paulo Costa 2

A Reference Architecture for Probabilistic Ontology Development
Richard Haberlin, Paulo Costa and Kathryn Laskey 10

Focused Belief Measures for Uncertainty Quantification in High Performance Semantic Analysis
Cliff Joslyn and Jesse Weaver 18

Recognizing and Countering Biases in Intelligence Analysis with TIACRITIS
Mihai Boicu, Gheorghe Tecuci, and Dorin Marcu 25

IAO-Intel: An Ontology of Information Artifacts in the Intelligence Domain
Barry Smith, Tatiana Malyuta, Ron Rudnicki, William Mandrick, David Salmen, Peter Morosoff, Danielle K. Duff, James Schoening and Kesny Parent 33

Managing Semantic Big Data for Intelligence
Anne-Claire Boury-Brisset 41

Context as a Cognitive Process: An Integrative Framework for Supporting Decision Making
Wayne Zachary, Andrew Rosoff, Stephen Read and Lynn Miller 48

Towards a Context-Aware Proactive Decision Support Framework
Benjamin Newsom, Ranjeev Mittu, Ciara Sibley and Myriam Abramson 56

Dynamic Data Relevance Estimation by Exploring Models (D2REEM)
H. Van Dyke Parunak 63

Data Analytics to Detect Evolving Money Laundering
Murad Mehmet and Duminda Wijesekera 71

Extraction of Semantic Activities from Twitter Data
Aleksey Panasyuk, Erik Blasch, Sue E. Kase and Liz Bowman 79

Situational Awareness from Social Media
Brian Ulicny, Jakub Moskal and Mieczyslaw M. Kokar 87

Towards a Cognitive System for Decision Support in Cyber Operations
Alessandro Oltramari, Christian Lebiere, Lowell Vizenor, Wen Zhu and Randall Dipert 94

Using a Semantic Approach to Cyber Impact Assessment
Alexandre de Barros Barreto, Paulo Cesar G Costa and Edgar Toshiro Yano 101

Analyzing Military Intelligence Using Interactive Semantic Queries
Rod Moten 109

Sketches, Views and Pattern-Based Reasoning
Ralph L. Wojtowicz 117

An Ontological Inference Driven Interactive Voice Recognition System
Mohammad Ababneh and Duminda Wijesekera 125

Fast Semantic Attribute-Role-Based Access Control (ARBAC)
Leo Obrst, Dru McCandless and David Ferrell 133

Supporting evacuation missions with ontology-based SPARQL federation
Audun Stolpe, Jonas Halvorsen and Bjørn Jervell Hansen 141

Navigation Assistance Framework for Emergencies
Paul Ngo and Duminda Wijesekera 149

Position Papers

Big Data for Combating Cyber Attacks
Terry Janssen and Nancy Grady 158

Hierarchical Decision Making
Matthew Lewis 162

Towards Context-Aware, Real Time and Autonomous Decision Making Using Information Aggregation and Network Analytics
Prithiviraj Dasgupta and Sanjukta Bhowmick 166

Need for Community of Interest for Context in Applied Decision Making
Peter S. Morosoff 170

Technical Papers

Context Correlation Using Probabilistic Semantics

Setareh Rafatirad
George Mason University
Email: srafatir@gmu.edu

Kathryn Laskey
George Mason University
Email: klaskey@gmu.edu

Paulo Costa
George Mason University
Email: pcosta@gmu.edu

***Abstract*—We present an approach for recognizing high-level geo-temporal phenomena – referred as events/occurrences– from in-depth discovery of information, using geo-tagged photos, formal event models, and various context cues like weather, space, time, and people. Due to the relative availability of information, our approach automatically obtains a probabilistic measure of occurrence likelihood for the recognized geo-temporal phenomena. This measure, however, is not only used to find the best event among the merely possible candidates – witnessing the data (including photos), but it can also provide informative cues to human operators in the environments where uncertainty is involved in the existing knowledge.**

I. Introduction

Sensors have become one of the biggest contributors of BIG DATA datasets. Numerous datasets have been already generated in real-time with rich content, about various information. Mobile wireless devices with multiple sensors like camera and GPS, and internet connectivity, can continuously capture photos and record camera parameters, GPS location, and time. The availability of various web services like *MapMyRide* [1], and *Wunderground* [2], provides semantics like ride, and geo-temporal weather status logs, using the captured sensory data. Given that context data exists in massive volumes, an information management paradigm is needed to correlate the information and infer higher level semantics. We propose a technique that automatically correlates various information, and creates a context-aware event graph by combining event models with contextual information related to photos, sensor logs, heterogeneous data sources, and web services. Our technique automatically computes the occurrence-likelihood for the event nodes in the output graph – referred as plausibility measure that provides informative cues to human operators in uncertain environments to make better decisions. Note that this work provides a holistic view of the high-level events witnessed by a dataset; further cause-effect decision-making using the output of this stage in out of the scope of this paper.

Events, in general, are structured and their subevents have relatively more expressive power [13]. In this work, an event model (or event ontology) provides a multi-granular conceptual description, i.e., it provides conceptual hierarchy in multiple levels using containment event-event relationships e.g., *subevent-of*, and *subClassOf*. In addition, event types can have multiple instances; instance events are contextual, and they should be augmented with context cues (like place, time, weather). This makes instance events more expressive than event types. Augmenting an instance event with context cues adapts a concept to multiple contextual descriptions (e.g., event type *visit-landmark* may have two instances; one instance associated with *World War II Memorial* and the other to *Washington Monument*). Consider the following example: A person takes a photograph at an airport less than 1 hour after his flight arrives. To explain this photograph, we first need the background knowledge about the events that generally occur in the domain of a trip. These semantics can only come from an event-ontology that provides the vocabulary for event/entity and event relationships related to a domain. An event-ontology allows explicit specification of models that could be modified using context information to provide very flexible models for high-level semantics of events. We refer to this modification as *Event Ontology Extension*. It constructs a more robust and refined version of an event-ontology either fully or semi-automatically. Secondly, given the uncertain nature of sensory data (like GPS that is not always accurate), the event type witnessed by the available context data is not decisive; in the above example, the event might either be *rent a car*, or *baggage claim* that are two possible conclusions — sometimes no single obvious explanation is available, but rather, several competing explanations exist and we must select the best one. In this work, reasoning from a set of incomplete information (observations) to the most related conclusion out of all possible ones (explanations) is performed through a ranking algorithm that incorporates the plausibility measure; this ranking process is used in *Event Ontology Extension*.

Problem Formulation: Every input photo has context information (timestamp, location, and camera parameters) and a user. Each photo belongs to a photo stream P of an event with a domain event model $O(V, E)$ –handcrafted by a group of domain experts– whose nodes V are event/entity classes, and edges E represent the relationships between the nodes. There is a bucket B of external data sources represented with a schema. The sources can be queried using the metadata of the input photographs and other available information, including the information about the associated user. Given P, B, O, and information associated to the user, how does one find the finest possible event tag that can be assigned to a photo or a group of similar photos in P?

Solution: We propose an Event Ontology Extension technique described as follows: select a relevant domain event model through the information related to both P and the user. Using P, B, O, and the user information, infer S – that consists of the best relevant subevent categories to P– where $S \subseteq V$. A member of S is the most plausible event category for a group of contextually-similar photos. For a group of similar photos c_j, a function f calculates the plausibility measure m_{ij}^p for every competing event candidate s_i: $f(s_i, c_j) = m_{ij}^p$; this measure indicates how much s_i is relevant to c_j such that

[1] http://www.mapmyride.com/
[2] http://www.wunderground.com/

$c_j \subset P$. Using the information from B, extend S with one or more augmented instances of S, and obtain expressive event tags T. An event tag $t_i^e \in T$ is a subevent of an event that either exists in O, or can be derived from O such that t_i^e is the finest subevent tag that can be assigned to a group of similar photos. If t_i^e is an assignable tag to any photo, and $t_i^e \notin O$, we intend to extend O by adding t_i^e to O such that the constraints governing O are preserved. The output is an extension to O that is referred as O_r (see fig 1). We argue that attribute values related to an inferred event need to be obtained, refined, and validated as much as possible to create very expressive and reliable metadata. Fig 3 depicts the processing components of our proposed approach. We used semantics such as spatiotemporal attributes/constraints of events, subevent structure, and spatiotemporal proximity. In contrast to machine learning approaches that are limited to the training data set and require an extensive amount of annotation, we propose a technique in which existing knowledge sources are modified and expanded with context information in external data sources including public data sources (like public event/weather directories, local business databases), and digital media archives (like photographs). With this knowledge expansion, new infrastructures are constructed to serve relevant data to communities. Event tags are propagated with event title, place information (like city, category, place name), time, weather, etc. Our proposed technique provides two unique key benefits as follows: *1)* A sufficiently flexible structure to express context attributes for events such that the attributes are not hardwired to events, but rather they are discovered on the fly. This feature does not limit our approach to a single data set; *2)* leveraging context data across multiple sources could facilitate building a consistent, unambiguous knowledge base.

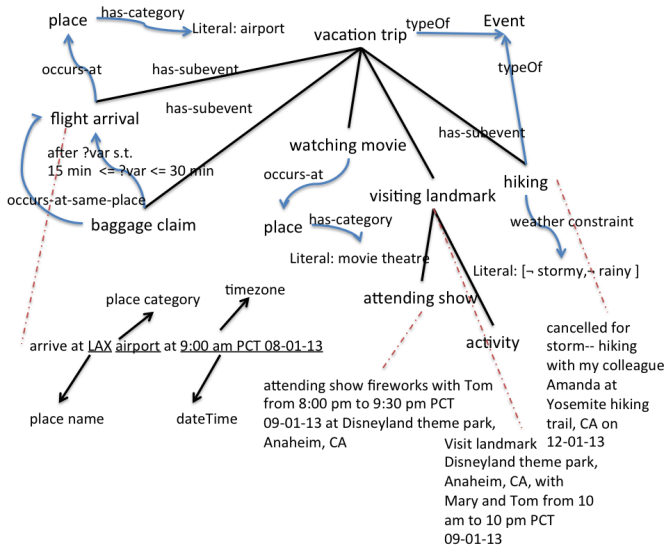


Fig. 1. An example of an event model being extended with contextually propagated instances.

Some of the main challenges of this work are: *a)* collecting and correlating information from various sources – we need a general mechanism that automatically queries sources and represents the output; *b)* a validation mechanism to ensure the coherency of the obtained data; *c)* currently, publicly available benchmark data sets such as those offered by TRECVid do not suit the purpose of this research (they deal with low level events i.e.,activities). However, higher-level events have relatively more contextual characteristics; *d)* according to the useful properties of photos, relevant event categories in the model must be discovered. This paper is organized as follows: in section II, we review the prior art that use context and event models for annotating photographs; in section III and IV, we explain our solution strategy; this is followed by section V that demonstrates our experiments, and section VI which is the conclusion.

II. State Of Art

The important role of context is emphasized in [9]. Context information and ontological event models are used in conjunction by [16], [6]. Cao et al. present an approach for event recognition in image collections using image timestamp, location, and a compact ontology of events and scenes [4]; this work, does not support subevent structure. Liu et al. reports a framework that converts each event description from existing event directories (like Last.fm) into an event ontology that is a minimal core model for any general event [11]. This approach is not flexible to describe domain events (like *trip*) and their subevent structure. Paniagua et al. propose an approach that builds an event hierarchy using the contextual information of a photo based on moving away from routine locations, and string analysis of English album titles (annotated by people) for public web albums in Picasaweb [12]. The limitations of this approach are: 1) human-induced tags are noisy, and 2) subevent relationship is more than just spatiotemporal containment. For instance, albeit a *car accident* may occur in the spatiotemporal extent of a *trip*, it is not part of the subevent-structure of the *trip*. According to [3], events form a hierarchical narrative structure that is connected by causal, temporal, spatial and subevent relations. If these aspects are carefully modeled, they can be used to create a descriptive knowledge base for interpreting multimedia data. In [14], a mechanism is proposed that exploits context sources in conjunction with subevent-structure of an event — this structure is modeled in a domain event ontology. The limitation of this approach is no matter how much an event category is relevant to a group of photos in a photo stream, it is used in photo annotation; as a result, the quality of annotation degrades.

III. Event Ontology Extension

Photo's incomplete information can be improved if combined with the information related to a group of similar photos. In this work, two images are similar if they belong to the same event type. Partitioning a photo stream of an event based on the context of its digital photographs can create separate subevent boundaries for its photos [5]. An event is a spatiotemporal entity [7]. In addition, optical camera parameters (CP) in photos provide useful information related to the environment (like *outdoor*) at which an event occurs [15]. We used a clustering that partitions photos hierarchically based on their timestamp, location, and CP. We used single linkage clustering and Euclidean distance in our clustering technique. However, one can use other approaches and refine the results. We present the observations (i.e., photos/clusters) with a set of descriptors – a cluster consists of a group of contextually similar photos. In this section, we show that it is feasible to go from a set of

descriptors D to the best subevent category, when the following conditions are satisfied: (a) the descriptors in D are consistent among themselves, (b) the descriptors in D satisfy subevent categories, (c) axioms of a subevent category are consistently formulated in an event ontology, and (d) the inferred subevent categories are sound and complete.

A. EVENT MODEL

We use a basic derivation of E* model [8] as our core event model, to specify the general relationships between events and entities. Specifically, we utilized the relationships *subeventOf*, which specifies the event structure and event containment. The expression e_1 $subeventOf$ e_2 indicates that e_1 occurs within the spatiotemporal bounds of e_2, and e_1 is part of the regular structure of e_2. Additionally, we used the spatiotemporal relationships like *occurs-during* and *occurs-at* to specify the space and time properties of an event. The time and space model that we used in this work is mostly derived from E* model. The relationship *participant* is used to describe the presence of a person in an event. We use the relationships *co-occurring-with*, and *co-located-with*, *spatially-near*, *temporal-overlap* , *before*, and *after* to describe the spatiotemporal neighborhood of an event. The relationship *same-as* between two events, makes them equivalent entities. Also, we used several other relationships to describe additional constraints about events (e.g., e_1 has-ambient-constraint A, and A has-value *indoor*). Moreover, to express a certain group of temporal constraints, we utilized some of Linear Temporal Logic, Metric Temporal Logic, and Real-Time Temporal Logic formulas [10], [2]. These formulas are a combination of the classical operators $\wedge$ (conjunction) , $\vee$ (disjunction) , implication ($\rightarrow$) , Allen's calculus [1], $\Box$ operator, $\Diamond$ operator, linear constraints, and distance functions; they are used to model complex relative temporal properties. For instance constraint $\Box_{[t_1,t_2]}(e_1 \rightarrow \Diamond_{[t_2,t_2+1800]} e_2 \wedge \tilde{\mathcal{D}}(e_2) \leq 1800)$ states that e_2 eventually happens within 1800 seconds after e_1 and that e_2 lasts less than or equal to 1800 seconds. We developed a language $\mathcal{L}$ with a syntax and grammar as an extension to OWL to embrace complex temporal formulas. Further, we extended the language to support a combination of classical propositional operators, linear spatial constraints, and spatial distance functions which can not be expressed in OWL; equation $f_{eucDist}(e_1, e_2, @ \leq 100)$ shows a relative spatial constraint in $\mathcal{L}$, which states the event e_1 occurs at most 100 meters away from the place at which event e_2 occurs.

Domain Event Model: A domain event ontology provides specialized taxonomy for a certain domain like *trip*, see fig 2. The *Miscellaneous* subevent category in this model is used to annotate the photos that are not matched with any other category. The general vocabulary in a core event model is reused in a domain event ontology. For instance, *Parking* in fig 2, is a *subClassOf* of *Occurrent* (or event) concept in the core event ontology. Also, relationships like *subeventOf* are reused from the core event ontology. We assume that domain event ontologies are handcrafted by a group of domain experts.

B. DESCRIPTOR REPRESENTATION MODEL

We represent a descriptor using the schema in script $\{type_d : value_d, confidence_d : val\}$, in which $type_d$, $value_d$, and val indicate the type, value,and certainty (between 0 and 1) of the descriptor, respectively. For instance, the descriptor $\{Flash : `off`, confidence : 1.0\}$ for a photo, states that the flash was off when the photo was captured with 100% certainty. Photo and cluster descriptors follow the same representation model, however the rules for computing the value of $confidence_d$ are different. We will describe these rules in the following paragraphs. The descriptor model of a cluster includes two fields in addition to that of a photo: plausibility-weight ≥ 0 , and implausibility-weight < 0. Later, we will explain the usage of these fields. All descriptors are either *direct* or *derived*. For photo descriptors, by convention, we assume that a direct descriptor is straightly extracted from the EXIF metadata of a photo, and its confidence is 1, as in the above example. The direct descriptors that we used in this paper are related to time, location, and optical parameters of photos like *GPSLatitude* ,*GPSLongitude* , *Orientation*, *Timestamp*, and *ExposureTime*. For a derived descriptor like $\{sceneType : `indoor`, confidence : 0.6\}$, the descriptor value 'indoor' is computed using direct descriptors like *Flash*, through a sequence of computations that extract information from a bucket of data sources. Some of these descriptors are *PlaceCategory*[3], *Distance*[4], and *HoursOfOperation*[5]. The confidence score is obtained from the processing unit used to compute the descriptor value — we developed several information retrieval algorithms for this purpose,in addition to the existing tools in our lab [15]. If a descriptor value is directly extracted from an external data source, $confidence_d$ is equal to 1. Direct descriptors of a cluster must represent all photos contained in it; some of these descriptors represent *boundingbox*, *time-interval*, and *size* of the cluster. The confidence value for direct descriptors is equal to 1, for instance, in the descriptor $\{size : 5, confidence_d : 1.0\}$ that indicates the number of photos in a cluster, $confidence_d$ is equal to 1.

Given a photo p_i in a photo stream P, and the cluster c that groups p_i with the most similar photos in P, a processing unit produces the descriptors of c using the descriptors of the photos in c, and more importantly, this process is guided by the descriptors of p_i. Every photo in c must support every *derived* descriptor of p_i; such cluster is referred as a *sound cluster* for p_i, and the *derived* descriptors for c are represented by the distinct union of the *derived* descriptors of the photos in c. For a derived cluster descriptor d, the value of $confidence_d$ is calculated using the formula in equation 1, in which $|c|$ is the size of the cluster, p_j is every photo in c that is represented by d, and $g(p_j, d)$ gives the confidence value of d in p_j. To find a sound cluster for a photo, the hierarchical structure that is produced by the *clustering* unit, is traversed using depth-first search — the halting condition for this navigation, if no sound cluster was found, is when current cluster is a leaf node.

$$confidence_d = \frac{1}{|c|} \times \sum f(p_j, d) \tag{1}$$

Descriptor Consistency: Consistency among a set of descriptors is a mandatory condition to infer the best possible conclusion from it. In this work, consistency must exist among the descriptors of a photo as well as the descriptors of a cluster, using entailment rules described below. *(a)* $v_i \rightarrow v_k$: if v_i implies v_k, then the rules for v_k must also be applied to v_i. This

[3]The category of the nearest local business to the coordinates of a photo.
[4]The distance of a local business to the coordinates of a photo.
[5]The hours during which a local business is open.

is referred as *transitive entailment rule*. For instance, suppose a photo/cluster has the following description, $'outdoorSeating : true' ; 'sceneType : outdoor'; 'weatherCondition : storm'$, which implies that the nearest local business (e.g. restaurant) to the photo/cluster, offers *outdoorSeating*, and the weather was stormy when the photo(s) were captured. Given the sequence of rules below,

$$outdoorSeating \wedge outdoor \rightarrow fineWeather,$$
$$fineWeather \rightarrow \neg storm$$

rule 2 is entailed that indicates an inconsistency among the descriptors of a photo/cluster.

$$outdoorSeating \wedge outdoor \rightarrow \neg storm \quad (2)$$

(b) $v_i \rightarrow func_{remove}(v_k)$: v_i implies removing the descriptor v_k. This is referred as a *deterministic entailment rule*.

(c) $v_i \wedge v_k \rightarrow truth\ value$: rules of this type are referred as *non-deterministic entailment rules* in which the inconsistency is expressed by a false truth value e.g. $closeShot \wedge landscape \rightarrow false$. In that case, further decisions on keeping,modifying, or discarding either of the descriptors v_i or v_k will be based on the confidence value assigned to each descriptor — this operation is referred as *update*, which is executed when an inconsistency occurs between two candidate descriptors. The following rules are used by this process: (a) for two descriptors with the same type, the descriptor with lower confidence score is discarded, (b) for two descriptors with different types, the one with lower confidence score gets modified until the descriptors are consistent. The modification is defined as either *negation* or *expansion* within the search space. In case of negation, e.g. $\neg outdoor \rightarrow indoor$, the confidence value for *indoor* descriptor is calculated by subtracting the confidence value of *outdoor* descriptor from 1. An example of expansion is increasing a window size to discover more local businesses near a location. To avoid falling inside an infinite loop, we limit the count of negation, and the size of search space during expansion, by a threshold. We assign *null* to the descriptor that has already reached a threshold and is still inconsistent. *null* is universally consistent with any descriptor. The vocabulary that is used to model the descriptors for a photo/cluster is taken from the vocabulary that is specified in the core event model.

C. DATA SOURCES

We represent each data source with a declarative schema, by using the vocabulary of the core event model. This schema indicates the type of source output. In addition, it specifies what type of the input attributes a source needs, to deliver the output. Data sources are queried using the SPARQL language[6]. A query is constructed automatically using the schema of data sources, and the available information. Simply put, a source is selected if its input attributes match the available information I. At every iteration, I is incrementally updated with new data that is delivered by a source. The next source is selected if its input attributes are included in I. This process continues until no more source with matching attributes is left in the bucket B.

[6]http://www.w3.org/TR/rdf-sparql-query/

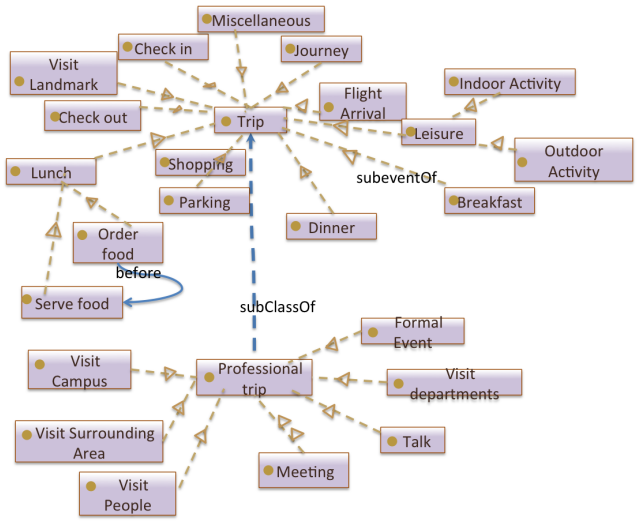


Fig. 2. An event ontology for the domain *professional trip*.

D. EVENT INFERENCE

From a set of consistent cluster descriptors (*observations*), we developed a *context discovery* algorithm to infer the most plausible subevent category described in a domain event ontology. This algorithm, uses the domain event model, which is a graph; we represent this graph with the notation $O(V, E)$ in which V includes event classes, and E includes event relationships. Traversing the event graph O starts with the root of hierarchical subevent structure. The algorithm visits event candidates in E through some of the relationships in E like *subeventOf*, *co-occurring-with*, *co-located-with*, *spatially-near*, *temporal-overlap*, *before*, and *after* — these relationships help to reach other event candidates that are in the spatiotemporal neighborhood of an event. An expandable list, referred as L_v, is constructed from E, to maintain the visited event/subevent nodes during an iteration i — if an event is added to L_v, it cannot be processed again during the extent of i. At the end of each iteration, L_v is cleared. In every iteration, the best subevent category is inferred through a ranking process, from a set of consistent observations.

To find the most plausible subevent category, we introduce *Measure of Plausibility* (m_{ij}^p) to rank event candidates. This measure is computed using two parameters (1) granularity score (w_g), and (2) plausibility score (w_{AX}). w_g is equivalent to the level of the event in the subevent hierarchy in the domain event ontology. To compute w_{AX}, we used 'plausibility-weight' (w^+) and 'implausibility-weight' (w^-) which are two fields of a cluster descriptor. The value of w^+ is equal to the confidence value assigned to a descriptor, and the value of w^- is equal to $-w^+$. If a descriptor could not be mapped to any event constraint, w_{AX} remains unchanged. If a descriptor with $w^+ = \alpha$ satisfies an event constraint, then w^+ is added to w_{AX}, otherwise, w^- is added to w_{AX} (i.e., $w_{AX} = w_{AX} - \alpha$). The only exception is for the cluster descriptors *time-interval* and *boundingbox*; if either one of these descriptors satisfies an explanation, then $w^+ = 1$; in the opposite case, $w^- \leq -100$ — when a cluster has no overlap with the spatiotemporal extent of an event s_i, $w^- \leq -100$ makes s_i the least plausible

candidate in the ranking. According to the formula in III-D, w_{AX} also depends on the fraction of satisfied event constraints; N is the total number of constraints for an event candidate.

$$w_{AX} = \frac{1}{N} \sum w^j_{AX} \quad , 1 \leq j \leq N \tag{3}$$

Finally, we use the following instructions to compare two event candidates e_1 and e_2: when e_1 is subsumed by e_2, m^p_{ij} for each event candidate is normalized using the formula in equation 4, in which $e_i \equiv e_1$ and $e_j \equiv e_2$, otherwise, $e_i.m^p_{ij} = e_i.w_{AX}$. The candidate with the highest m^p_{ij} is the most plausible subevent category.

$$e_i.m^p_{ij} = \frac{e_i.w_{AX}}{\max(e_i.w_{AX}, e_j.w_{AX})} + \frac{e_i.w_g}{\max(e_i.w_g, e_j.w_g)} \tag{4}$$

When a subevent category is inferred from a set of observations, it will not be considered again as a candidate for the next set of observations. Event inference halts if no more subevent category is left to be inferred from the domain event ontology.

EXTENSION: The inferred subevent categories E' are refined with the context data extracted from data sources in the bucket B, through the refinement process. First, let us elaborate this process by introducing the notion of *seed event*, which is an instance of an inferred category in E', which is not yet augmented with information. An augmented seed-event is an expressive event tag. The seed-event is continuously refined with information from multiple sources.

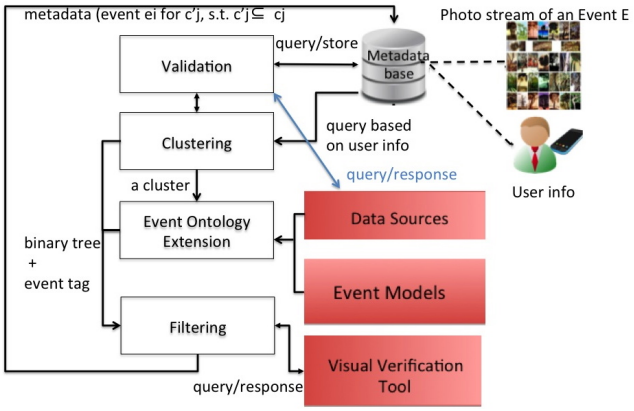


Fig. 3. The Big Picture. Photos and their metadata are stored in *photo-base* and *metadata-base* respectively. Using *user info*, including events' type, time, and space in a user's calendar, a photo stream is queried, and its metadata is passed to *Clustering*. In *Validation*, a set of consistent descriptors is obtained from the cluster that best represents an individual photo — the component *event inference* uses these descriptors in addition to a domain model that is selected according to *user info*. *Event Ontology Extension* propagates the most relevant subevent categories (to the input photo stream) with the information discovered from *Data Sources*, then extends the event structure (ontology) with the applicable propagated event instances (i.e., tags). The tags are validated (using data sources), and added to the event ontology – the extended event ontology is used in *filtering* that queries *visual concept verification tool*. In this stage given an event, irrelevant cluster branches are pruned. Next, for each matched cluster, less relevant photos to a subevent tag are filtered. The output is a set of photos labeled with some tags; these tags are then stored as new metadata for the photos. The remaining photos are tagged as *miscellaneous*.

Our extension algorithm uses a similar strategy as what we used in subsection III-C. The difference is, the attributes of a data source at each iteration is supplemented by the user information and the attributes of a seed-event (I) that is represented with the same schema that is described in the event ontology. Given a sequence of input attributes, if a data source returns an output-array of size K, then our algorithm creates K new instances of events with the same type as in the seed-event, and augments them with the information in the output-array. The augmented seed-events are added to I for the next iteration; I is constantly updated until all the event categories in E' are augmented, and/or there is no more data source (in the bucket B) to query. To avoid falling into an infinite loop of querying data sources, we set the following condition: a data source cannot be queried more than once for each seed-event. We defined some queries manually that are expressed through the relative spatiotemporal relationships in the event ontology, and the augmented seed-events; these queries are used to augment the seed-events with relative spatiotemporal properties. When a seed-event gets augmented with information, our technique validates the event tag by using the event constraints, augmented event attributes, and a sequence of entailment rules that specify the *cancel* status for an event. For instance, if the weather attribute for an event is *heavy rain*, and the weather constraint *fine weather* is defined for an event, then the status of the event tag becomes *canceled*. After the validation, event tags are added to the domain event ontology by extending event classes through *typeOf* relationship. This step produces an augmented event ontology that is the extended version of the prior model (see fig 1).

IV. FILTERING

Filtering is a two-step process; (1) redundant and irrelevant clusters are pruned from the hierarchical cluster structure produced by the *clustering* component, see fig 4-step-1. (2) filter redundant photos from the matched cluster, see fig 4-step-2. This is accomplished by applying the context and visual constraints of the expressive tag that is matched to the cluster. We used a concept verification tool[7] to verify the visual constraints of events using image features. This tool uses pyramids of color histogram and GIST features. Filtering operation is deeply guided by the expressive tags. During this operation, subevent relations are used for navigating the augmented event model.

V. EXPERIMENTAL EVALUATIONS

We focused on 3 domain scenarios vacation, professional trip, and wedding. We crawled Flickr, Picasaweb, and our lab data sets. We observed that many people store their personal photos according to events; accordingly, we collected the data sets based on time, space, and event types (like travel, conference, meeting, workshop, vacation, and wedding). We developed some crawlers to download about 700 albums of the day's featured photos; we crawled photo albums created since the year 2010 since most of the older collections did not contain geo-tagged photos. After 4 months, we collected 570 albums (about 60K photos) which had the required EXIF information containing location, timestamp, and optical camera parameters. We ignored the albums a) smaller than 20 photos, b) with non-English annotations. The average number of photos per album was 105. We used the albums from the most

[7] http://socrates.ics.uci.edu/Pictorria/public/demo

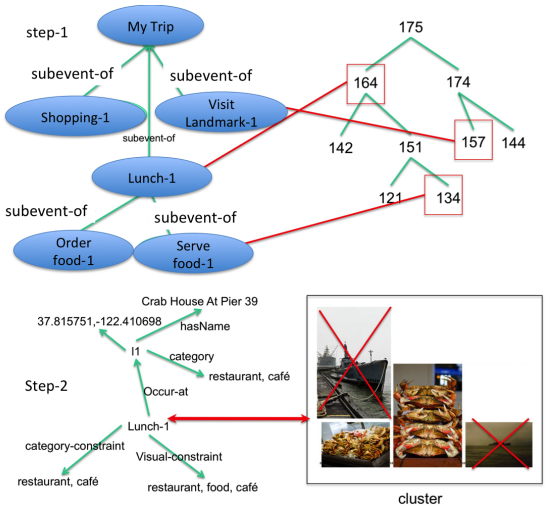


Fig. 4. Filtering Operation.

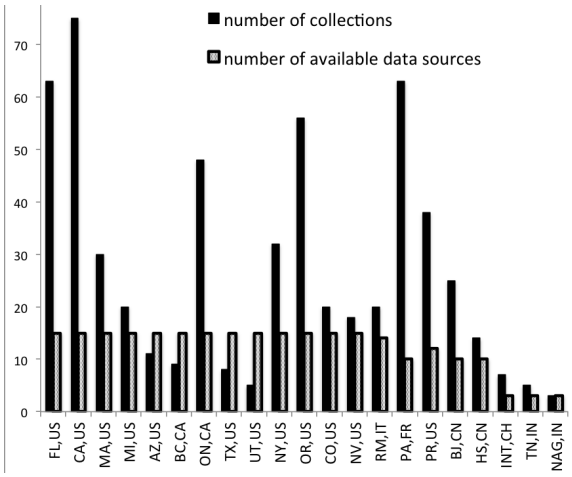


Fig. 5. Data set geographical distribution. The black bars show the number of albums in each geographic region, and the gray bars show the number of data sources that supported the corresponding geographic region.

active users based on the amount of user annotations, ending up with a collection of 20 users with heterogeneous photo albums in terms of time period and geographical sparseness. The geographic sparseness of albums ranged from being across continents, to cities of the same country/state (see fig 5). We noticed that data sources do not equally support all the geographic regions; e.g., only a small number of data sources supported the data sets captured inside India. The photos for vacation/professional-trip domains have higher temporal and geographical sparseness compared to photos related to wedding domain. The number of albums for vacation domain exceeds the other two.

Experimental Set-Up

We picked the 4 most active users (based on the amount of user annotation) from our non-lab, downloaded data set, and 2 most active users from our lab data set (based on the number of collections they own). As ground-truth for the lab data set, we asked the owners to annotate the photos using their personal experiences, and an event model that best describes the data

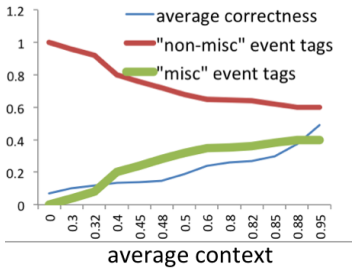


Fig. 6. Role of context in improving the correctness of event tags.

set, while providing them with three domain event models. For the non-lab data set, the ground truth provides a manual and subjective event labeling done by the very owner of the data set being unaware of the experiments. Because of the subjective nature of the non-lab data set, the event types that were not contained in the event domain ontology are replaced with event type *miscellaneous* that is an event type in every domain event ontology in this work. For each experiment, we compute standard information retrieval measures (precision, recall, and F1-measure), for the event types used in tags. In addition to that, we introduce a measure of correctness for event tags. The score is obtained based on multiple context cues. For instance, label *meeting with Tom Johnson at RA Sushi Japanese Restaurant in Broadway, San Diego, during time interval "blah" in a sunny day, in an outdoor environment*, specifies type of the event, its granularity in the subevent hierarchy, place, time, and environment condition. We developed an algorithm that evaluates each cue with a number in the range of 0 to 1 as follows: *1)* event type: wrong = 0, correct = 1, somehow correct = $\frac{L_p}{L_{TP}}$ such that L_p is the subevent-granularity level for a predicted tag and L_{TP} is the subevent granularity level for the true-positive tag (the predicted tag is the direct or indirect superevent of the true-positive tag i.e., $\frac{L_p}{L_{TP}} \leq 1$); *2)* place: includes place name, category and geographical region. If the place name is correct, score 1 is assigned and the other attributes will not be checked. Otherwise, 0 is assigned; for the category and/or geographical region if correct, score 1 is assigned, and 0 otherwise. The average of these values represent the score for place; *3)* for weather, optical, and visual constraint: wrong=0, correct =1, unsure = 0.5; *4)* time interval: if the predicted event tag occurs anytime during the true-positive event tag, 1 is the score, otherwise 0. The average of the above scores represents the correctness measure for a predicted event tag. We introduce *average correctness* of annotation that is calculated using the formula in equation 5, where w_j is the score for the j^{th} predicted tag, and L is the total number of expressive event tags detected by our approach.

$$\overline{correctness} = \frac{\sum_{j=1}^{L} w_j}{L} \; ; \; \overline{context} = 1 - \overline{Err} \qquad (5)$$

The metric $\overline{context}$ in equation 5 is used to measure the average context provided by data sources for annotating a photo stream; parameter $\overline{Err}$ is the average error related to the information provided by data sources used for annotating a photo stream ($0 \leq \overline{Err} \leq 1$); the following guidelines are applied automatically, to measure this value: *(a)* if the

information in a data source is related to the domain of a photo stream, but it is irrelevant to the context of the photo stream, assign error-score 1. For instance, data source *TripAdvisor* returns zero results related to *Things-To-Do* for the country at which a photo stream is created. Also, if a photo stream for a vacation trip does not include any picture taken in any landmark location, *TripAdvisor* does not provide any coverage; *(b)* assign error-score 0 if the type of a source is relevant as well as its data (i.e. non-empty results); *(c)* if the data from a relevant source is insufficient for a photo stream, assign error-score 0.5. For instance, only a subset of business venues in a region are listed in data source *Yelp*; as a result, the data source returns information for less than 30% of the photo stream; *(d)* for a data source, multiply the error-score by a fraction in which the numerator is the number of photos tagged using this data source, and the denominator is the size of the photo stream. Do this for all the sources and obtain the weighted average of the error-scores. The result is $\overline{Err}$. The implication of our result in fig 6 is as follows: while the correctness of event tags (for a photo stream of an event) peaks with the increase in $\overline{context}$, relatively, smaller percentage of photos are tagged using *non-miscellaneous* events, and larger percentage of photos are tagged using *miscellaneous* event. This means if the suitable event type for a group of photos does not exist in an event ontology, the photos are not tagged with an irrelevant *non-miscellaneous* event; instead, they are tagged with *miscellaneous* event which means *other*. The right side of the figure indicates that even though the number of miscellaneous and non-miscellaneous event tags does not change, the correctness is still increasing; this means that the tags get more expressive since more context cues are attached to them. The quality of annotations is increased when more context information is available. This shows that event ontology by itself is not as effective as augmented event ontology. We demonstrate three classes of experiments in table I. This table shows the average values (between 0 to 1) for the measure metrics discussed earlier (precision, recall, F1, $\overline{correctness}$). We use the work proposed in (Paniagua, 2012) as a baseline. It is based on space and time to detect event boundaries in conjunction with using English album descriptions. This baseline approach, with F1-measure about 0.6 and correctness of almost 0.56, illustrates that time and space are important parameters to detect event boundaries. On the other hand, the baseline approach is limited to using only spatiotemporal containment for detecting subevent hierarchy, it does not support other types of relationships among events (like co-occurring events, relative temporal relationships) and other semantic knowledge about the structure of events. Also, it requires human-induced tags which are noisy. For the second set of experiments, we use an event domain ontology without augmenting it with context information. This approach gives worse results since the context information is disregarded during detecting event boundaries. It provides the F1-measure of almost 0.32 and correctness of 0.13. Our last experiment leverages our proposed approach, and achieves F1-measure of about 0.85, and correctness of 0.82. Compared to our baseline approach, we obtain about 26% improvement in the quality of tags which is a very promising result.

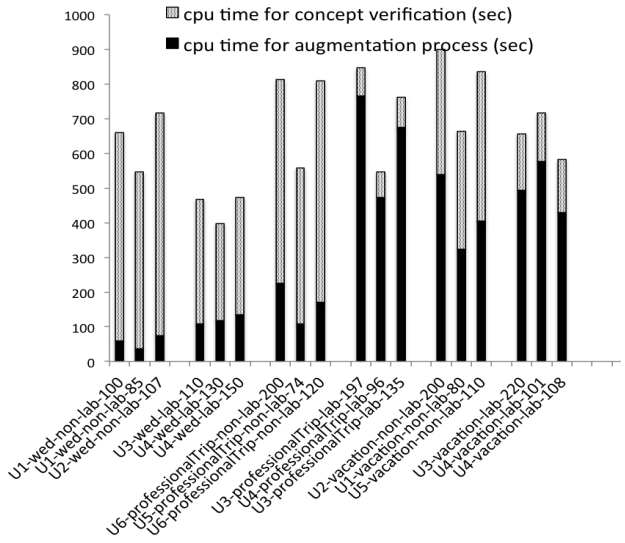


Fig. 7. CPU-Time for experimental data sets of the 5 most active users. Each data set is represented by its owner, domain type, source, and size. The domain *wed* implies *wedding* domain.

CPU-Performance

The running time for our proposed approach, and visual concept verification is shown in fig 7, which illustrates the results for data sets of two sources i.e., lab, and non-lab (including Flickr, and Picasaweb), and three event domains.

Cross-Domain Comparison : In general, we found smaller number of context sources for wedding data sets compared to the other two domains; as a result, the extension process exits relatively faster, and the running time for the concept verification process increases. We observed the correctness of event tags degrades when *Event Ontology Extension* process exists fast. This observation confirms the findings of fig 6.

Cross-Source Comparison: Within each domain, we compared the cpu-performance among lab and non-lab data sets; Event Ontology Extension exits relatively faster for non-lab data sets. The justification for this observation is that we could obtain user-related context like facebook events/check-ins from our lab users (U3, U4), but such information was missing in the case of non-lab data sets. This absence of information impacts wedding data sets the most, since the context information in the *wedding* scenario largely includes personal information such as guest list, and wedding schedule that are not publicly available on photo sharing websites. In *professionalTrip* scenario, this impact is smaller than *wedding*, and larger than *vacation*; the missing data is due to the lack of context information related to personal meetings, and conference schedules. In *vacation* scenario, data sources are mostly public; only a small portion of context information comes from the user-related context such as flight information,and facebook check-ins; therefore, we did not find a significant change in the cpu-time between lab and non-lab data sets.

VI. CONCLUSIONS

Our proposed technique addresses a broad range of research challenges to achieve a powerful event-based system that can adapt to different scenarios and applications like

Users		U1	U2	U3	U4	U5
baseline	prec	0.65	0.58	0.39	0.53	0.74
	recall	0.89	0.4	0.61	0.64	0.8
	f1	0.75	0.47	0.48	0.6	0.77
	corr	0.63	0.62	0.52	0.62	0.28
event ontology	prec	0.41	0.17	0.3	0.48	0.12
	recall	0.4	0.2	0.5	0.43	0.24
	f1	0.4	0.18	0.37	0.45	0.16
	corr	0.2	0.08	0.12	0.2	0.03
proposed	prec	0.74	0.83	0.95	0.92	0.88
	recall	0.91	0.93	0.88	0.7	0.97
	f1	0.81	0.88	0.91	0.79	0.92
	corr	0.8	0.75	0.85	0.79	0.9

TABLE I. RESULTS FOR AUTOMATIC PHOTO ANNOTATION FOR THE DATA SETS OWNED BY THE 5 MOST ACTIVE USERS.

those in intelligence community, multimedia applications, and emergency response. This is the starting step for combining complex models with *BIG DATA*.

REFERENCES

[1] J. F. Allen and G. Ferguson. Actions and events in interval temporal logic. In Journal of Logic and Computation, 1994.

[2] R. Alur and T. A. Henzinger. Logics and models of real time: A survey. In J. W. de Bakker, Cornelis Huizing, Willem P. de Roever, and Grzegorz Rozenberg, editors, REX Workshop, Springer, 1991.

[3] N. Brown. On the prevalence of event clusters in autobiographical memory. Social Cognition, 2005.

[4] L. Cao, J. Luo, H. Kautz, and T. Huang. Annotating collections of photos using hierarchical event and scene models. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE.

[5] M. Cooper, J. Foote, A. Girgensohn, and L. Wilcox. Temporal event clustering for digital photo collections. ACM Transactions on Multimedia Computing, Communications, and Applications, 2005.

[6] A. Fialho, R. Troncy, L. Hardman, C. Saathoff, and A. Scherp. What's on this evening? designing user support for event-based annotation and exploration of media. In 1st International Workshop on EVENTS-Recognising and tracking events on the Web and in real life, 2010.

[7] B. Gong, U. Westermann, S. Agaram, and R. Jain. Event discovery in multimedia reconnaissance data using spatio-temporal clustering. In Proc. of the AAAI Workshop on Event Extraction and Synthesis, 2006.

[8] A. Gupta and R. Jain. Managing event information: Modeling, retrieval, and applications. Synthesis Lectures on Data Management, 2011.

[9] R. Jain and P. Sinha. Content without context is meaningless. In Proceedings of the international conference on Multimedia. ACM, 2010.

[10] R. Koymans. Specifying real-time properties with metric temporal logic. In Real-Time Syst.,2(4), 1990.

[11] X. Liu, R. Troncy, and B. Huet. Finding media illustrating events. In Proceedings of the 1st ACM International Conference on Multimedia Retrieval. ACM, 2011.

[12] J. Paniagua, I. Tankoyeu, J. Stöttinger, and F. Giunchiglia. Indexing media by personal events. In Proceedings of the 2nd ACM International Conference on Multimedia Retrieval. ACM, 2012.

[13] S. Rafatirad, A. Gupta, and R. Jain. Event composition operators: Eco. In Proceedings of the 1st ACM international workshop on Events in multimedia. ACM, 2009.

[14] S. Rafatirad and R. Jain. Contextual augmentation of ontology for recognizing sub-events. In Semantic Computing (ICSC), 2011 Fifth IEEE International Conference. IEEE, 2011.

[15] P. Sinha and R. Jain. Classification and annotation of digital photos using optical context data. In CIVR, 2008.

[16] W. Viana, J. Bringel Filho, J. Gensel, M. Villanova-Oliver, and H. Martin. Photomap: from location and time to context-aware photo annotations. Journal of Location Based Services, 2008.

A Reference Architecture for Probabilistic Ontology Development

Richard J. Haberlin, Jr.
EMSolutions, Inc.
Arlington, Virginia
rjhaberlin@comcast.net

Paulo C. G. da Costa
Kathryn B. Laskey
Systems Engineering and Operations Research
George Mason University
Fairfax, Virginia
pcosta, klaskey @gmu.edu

***Abstract* - The use of ontologies is on the rise, as they facilitate interoperability and provide support for automation. Today, ontologies are popular for research in areas such as the Semantic Web, knowledge engineering, artificial intelligence and knowledge management. However, many real world problems in these disciplines are burdened by incomplete information and other sources of uncertainty which traditional ontologies cannot represent. Therefore, a means to incorporate uncertainty is a necessity. Probabilistic ontologies extend current ontology formalisms to provide support for representing and reasoning with uncertainty. Representation of uncertainty in real-world problems requires probabilistic ontologies, which integrate the inferential reasoning power of probabilistic representations with the first-order expressivity of ontologies. This paper introduces a systematic approach to probabilistic ontology development through a reference architecture which captures the evolution of a traditional ontology into a probabilistic ontology implementation for real-world problems. The Reference Architecture for Probabilistic Ontology Development catalogues and defines the processes and artifacts necessary for the development, implementation and evaluation of explicit, logical and defensible probabilistic ontologies developed for knowledge-sharing and reuse in a given domain.**

Keywords—probabilistic ontology, knowledge engineering, reference architecture

I. INTRODUCTION

The Reference Architecture for Probabilistic Ontology Development (RAPOD) presents a compilation of components required for probabilistic ontology development and therefore facilitates design, implementation, and support processes without rigid adherence to a particular set of tools. The Department of Defense (DOD) defines a Reference Architecture as:

> *"… an authoritative source of information about a specific subject area that guides and constrains the instantiations of multiple architectures and solutions*[1]*."*

Common throughout the literature on reference architectures is the idea of serving as a blueprint for architects to develop specific solution architectures within a defined domain [1] [2]. As the blueprint, it serves as a template for software development, defining integral components and their relationships, thereby reducing development time and project risk. Further, it standardizes language among participants, provides consistency of development within the domain, provides a reference for evaluation, and establishes specifications and patterns [1].

A. Background

Development of the RAPOD provides synergy of effort within the Semantic Technology (ST) community by identifying concepts, processes, languages, theories and tools for designing and maintaining probabilistic ontologies. Presently, ontological engineering facilitates the development of explicit, logical and defensible ontologies for knowledge-sharing and reuse. A similar pragmatics in the form of the Probabilistic Ontology Development Methodology has been produced for probabilistic ontologies and is described in [3]. The RAPOD facilitates synergy of effort between multiple disciplines including probabilists, logicians, decision analysts and computer scientists. It describes each of the components required for a functional probabilistic ontology and their interrelationships, and defines the criteria to be satisfied by any set of selected tools and methods using a Unified Process-inspired methodology.

B. Scope

The RAPOD spans the knowledge, processes, models, and tools necessary for engineering probabilistic ontologies at a high level of abstraction. Through decomposition or aggregation of existing methodologies, it provides universal techniques and a generalized framework for the fundamental components needed to construct probabilistic ontologies from conceptualization to operation through multiple tasks, including:

- Model conceptualization and framing
- Ontology development through elicitation and ontological learning
- Probability incorporation through iterative decomposition

There are many participants involved in realizing an operational probabilistic ontology. The Stakeholder Decision Maker (DM), Subject-Matter Expert (SME) and Probabilistic

Ontology Developer coordinate to instantiate a collection of concepts and tools for development and implementation from existing and proposed ontological and probabilistic ontological engineering methodologies, providing a single collection of knowledge to solve a domain-specific problem. Their solution is defined as a domain-specific architecture that may be reused for comparable problems in similar domain contexts.

C. Model Implementation and Viewpoint

The concept behind the RAPOD is to establish intellectual control of the probabilistic ontology (PO) model, stimulate reuse, and provide a basis for development through instantiation of a particular set of tools the developer will utilize to design and implement complex probabilistic ontologies for a particular domain [4]. Intellectual control establishes common semantics and allows consistent integration of new system components by anticipating their inclusion from design. Reuse is a prime tenet of ontological engineering and is enabled through identification of common components and relationships. Further, a well-defined and properly architected PO may be reused entirely through spiral modification to incorporate additional knowledge or relationships. Most importantly, the architecture serves as a blueprint for the PO Developer and a clear mechanism between him and the Stakeholder Decision Maker. The architecture allows individuals, teams, and organizations to communicate objectives, requirements, constraints, components and relationships with a common vocabulary and understanding of the objective. Ontological engineering, and probabilistic ontological development, may be completed by several different methodologies depending on the context and domain of the problem. Therefore, the RAPOD provides ready access to tools, techniques, and procedures that have proven successful in the past. The RAPOD also exposes synergies in algorithms, heuristics and model use between ontological and probabilistic ontological engineering. Through careful selection of tools with common parameters, the final model is more intuitive. The viewpoint of this reference architecture is that of the Probabilistic Ontology Developer in support of a Stakeholder Decision Maker desiring decision support for a defined area of interest.

II. Reference Architecture for Probabilistic Ontology Development

The Reference Architecture for Probabilistic Ontology Development facilitates PO development and reuse by providing a template from which multiple PO solutions to similar problems may be constructed. The output of the RAPOD is a domain and problem-type specific architecture that may be used to develop POs for similar problems. Reusable architectures provide a shortcut to future development by identifying inputs, methodologies, and support artifacts that have previously produced successful solutions within the domain.

In each of its three layers, the RAPOD identifies processes and artifacts necessary for the construction of a probabilistic ontology without specification to particular tools. Working with the stakeholders, the PO Developer selects individual component solutions that suit the problem-type and domain. Specification of a set of tools for each component instantiates an architecture that is used to develop the PO. Figure 1 provides an overview of the RAPOD, discussed in detail below.

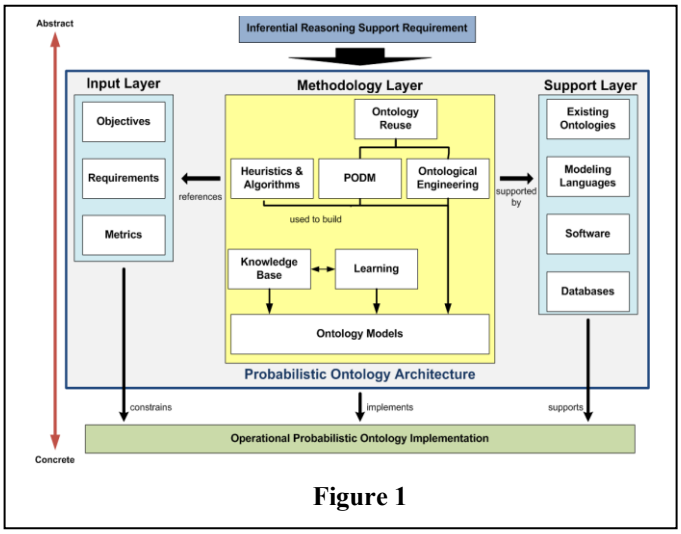


Figure 1

The Reference Architecture for Probabilistic Ontology Development shown in Figure 1 illustrates the scope of the reference architecture from abstract to concrete. At the top of the illustration is the most abstract conceptualization defined as a problem or objective by the Stakeholder Decision Maker that requires implementation of a probabilistic ontology. For example, a military commander may be charged with creating a decision support system that assists in the determination of an opposing force given limited sensor information. A Naval application example is given in [3]. The base of the illustration represents the operational implementation of the probabilistic ontology to provide inferential reasoning support. Between lies the probabilistic ontology architecture, which translates the conceptualization into a blueprint for development. The probabilistic ontology architecture is comprised of three interacting layers, which group and characterize similar functionality: the Input Layer, Methodology Layer, and Support Layer. These and their relationships are described in the following subsections.

A. Input Layer

The Input Layer defines external influences on the probabilistic ontology and is referenced by components of the Methodology Layer. It contains those components expected to provide detail on the purpose of the PO and its bounding constraints in the form of system requirements. Population of the Input Layer occurs primarily during the early stages of the development process during which the Stakeholder Decision Maker and PO Developer work closely to identify the objective of the model, expectations of its performance, and resource restrictions. Parameters specified in the Input Layer will constrain the operational implementation.

1) Objectives

The objectives hierarchy contains a representation of performance, cost and schedule attributes that determine the value of the system, with an over-arching Objective Statement that captures its primary intent [5]. Objectives state the overall intent of the project in short, clear, descriptive phrases. They

are defined by the Stakeholder DM to bound the scope of the final product and set expectations. These are often described in the following form [6]:

To Action + Object + Qualifying phrase

For a probabilistic ontology model, applicable categories of objectives may include: performance, reliability, compatibility, adaptability, and flexibility. Further descriptions of these and other categories may be found in Armstrong [6]. Choosing the correct objectives ensures that the desired problem is solved and that the PO Developer and Decision Maker have clearly communicated. The entire project is best focused through a Top-level Objective Statement.

2) Requirements

Requirements define the system to be implemented in terms of its behaviors, applications, constraints, properties, and attributes. The systems engineering literature on requirements elicitation and development is rich, but there is consensus that no single methodology exists for requirements engineering [7] [8]. In general, requirements elicitation approaches may be categorized as structured or unstructured [8] using a combination of strategies depending on the scope of the system under development and the participation commitment of the Stakeholder Decision Maker.

Requirements are elicited from the Stakeholder Decision Maker and SMEs through an iterative process that generally includes objective setting, background knowledge acquisition, knowledge organization, and requirements collection as introduced by Kotonya and Sommerville [7]. Grady categorizes three strategies for requirements analysis: structured analysis, cloning, and freestyle [8]. Using one or more of these strategies and concentrating on the four tasks above will lead to identification of appropriate requirements to satisfy valid model development. There is inefficiency and risk involved in the unstructured methods as there is nothing to prevent duplicative work, incompleteness, conflicts and misdirection.

3) Metrics

Metrics are used to describe parameters, Measures of Performance (MOP) and Measures of Effectiveness (MOE) that characterize the criteria against which the fielded system is to be evaluated. Green defines a hierarchy of effectiveness measures that follows the system of systems concept [9]. The following definitions are adapted from those offered by Green to accommodate the PO development process:

Measures of Effectiveness. A measure of system performance within its intended environment (e.g. overall system effectiveness).

Measures of Performance. A measure of one attribute of system behavior derived from its parameters (e.g. probability of correct identification).

Parameters. Properties or characteristics whose values determine system behavior (e.g. error rate).

Armstrong [6] opines that useful metrics take quantifiable form with both a clear definition of the measure and its associated units. They must also be mission-oriented, discriminatory, sensitive, and inclusive [9]. In all cases, appropriate metrics depend on the system under development and its ultimate purpose (objectives).

B. Methodology Layer

The Methodology Layer contains the heart of the probabilistic ontology development process including the Probabilistic Ontology Development Methodology that allows creation of a specific probabilistic ontology implementation to support the requirements of a Stakeholder Decision Maker. The Methodology Layer references information gathered in the Input Layer and is assembled using components and tools from the Support Layer. Its individual components are introduced below.

1) Probabilistic Ontology Development Methodology

The Probabilistic Ontology Development Methodology provides specific activities and tasks that evolve Stakeholder Decision Maker requirements into an ontology that is probabilistically-integrated, a probabilistic ontology. The activities of the Probabilistic Ontology Development Methodology are shown in the below activity diagram (Figure 2) and further detailed in [3]. These activities fit well within both Waterfall and Spiral Development Life Cycle processes where in Spiral Development iteration is explicitly anticipated.

Completion of the PODM activities and tasks establishes a framed solution to a specific inferential reasoning problem grounded in an inclusive ontology representing its entities and incorporating probability to represent uncertainty.

2) Ontological Engineering

In Gomez-Perez et al, ontological engineering is defined as the activities that concern the ontology development process, life cycle, construction methodologies and tools [10]. While traditional ontological engineering methods ensure that ontologies are explicit, logical and defensible, these methods provide insufficient support for the complexity of probabilistic ontology development, as discussed above. A systematic approach to PO development is needed that addresses the evolution of requirements into an ontology that is probabilistically integrated. The underlying ontology may be engineered by many methods; but ultimately each

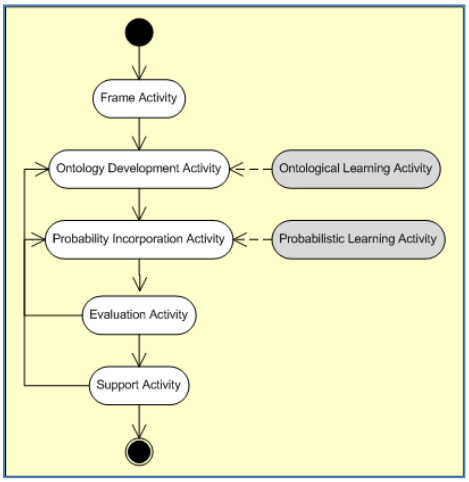


Figure 2

methodology provides a structured means to produce ontologies from conceptualization to implementation. Some principal design criteria must always be considered: clarity, coherence, extendibility, minimal encoding bias, and minimal ontological commitment [11].

3) Ontology Reuse

There are two types of ontology reuse: re-engineering and merging. Ontology re-engineering involves transforming the conceptual model of an implemented ontology into another conceptual model [10]. On the other hand, ontology merging uses information captured about one or more domains of interest in the creation of a new ontology. Therefore, model reuse is the process by which available knowledge and conceptual models are used as input to generate new models, in this case ontologies and probabilistic ontologies. Ontology development is a complex and labor-intensive task. The potential for reuse is an identified strength of ontologies and allows expansion of existing knowledge bases by capitalizing on previous research and development [10][11][12][13][14]. The literature liberally addresses the concept of ontology reuse, but there is little guidance offered for selection of methods for merging and/or integration. Integration of similar tasks and the addition of tasks emphasizing utility of existing ontologies expand the basic process of ontological engineering to make use of ever-expanding online ontology resources. Before beginning construction of a new ontology, it is useful to research existing ontologies in related domains to be reused and/or extended for the current problem. The ST community is actively expanding free access to the growing body of ontological knowledge, as discussed below.

4) Heuristics and Algorithms

Generally, a heuristic is an experience-based technique for problem solving, learning, and discovery and an algorithm is a stepwise procedure for calculation of a problem solution. Heuristics and algorithms are used to express relationships between classes within ontologies and probabilistic ontologies in order to constrain the models. For example, the heuristic "A weapon is cued by a single sensor" gives a plain-language description of a relationship in which each weapon is assigned a single sensor, but sensors may be assigned multiple weapons. This plain language description captures the machine-readable cardinality statement of ∞…1 in a format understandable by the entire development group, including the Stakeholder Decision Maker and SMEs. Heuristics and algorithms are captured as part of the PODM as described in [3].

5) Learning

Currently, ontology development is a labor-intensive, manual process. However, the need for greater automation features has been recognized and is a focus of the ST community. The PODM has integration points primed for future expansion in the areas of Ontological Learning and Probabilistic Learning. These two functions assist the modeler in ontology creation and elicitation of probabilities for the probabilistic relationships used for inferential reasoning.

a) Ontological Learning

Ontological learning is the process of extracting relevant classes, properties and relationships from a given data set, in this case to reduce effort in development of an ontology which will be developed into a probabilistic ontology. Buitelar et al. identified innovative aspects of ontology learning that set it apart from traditional knowledge acquisition [15]:

- It is inherently multidisciplinary due to its strong connection with the Semantic Web, which has attracted researchers from a very broad variety of disciplines: knowledge representation, logic, philosophy, databases, machine learning, natural language processing, image processing, etc.
- It is primarily concerned with knowledge acquisition from and for Web content and is moving away from small and homogeneous data collections.
- It is rapidly adapting the rigorous evaluation methods that are central to most machine learning work.

Through application of ontological learning, both the process of developing a probabilistic ontology and the development risk may be reduced.

Sowa defines three types of ontologies: a formal ontology which is a conceptualization whose categories are distinguished by axioms and definitions and are stated in logic to support inference and computation, a prototype-based ontology in which categories are formed by collecting instances extensionally, and a terminological ontology which describes concepts by labels and synonyms without axiomatic grounding [16]. Ontological learning in support of inferential reasoning is concerned primarily with developing the latter two categories for the specified domain of interest. The various sources used for ontology elicitation may include databases, documents, and taxonomies. As ontologies are typically hierarchically arranged, the primary means for ontological learning is through clustering. In this method, using a suitable clustering algorithm, a semantic distance is measured between terms and the nearest terms are clustered and formed into a prototype-based ontology. Ontological learning may also be accomplished through pattern matching using a co-occurrence matrix or bootstrapping from a seed lexicon that is extended by measuring similarity.

The above methods are all primarily focused on learning ontologies from plain text corpuses. Recent work includes extracting ontologies from non-text formats including relational databases, structured knowledge bases, and the Semantic Web. Albarrak developed an extensible framework for generating ontologies from Relational Database (RDB) and Object-Relational Database (ORDB) data models [17]. Li et al. introduce a novel set of 12 learning rules that build a complete OWL ontology of classes, properties, characteristics, cardinality and instances [18]. A database analyzer extracts key information from the relational database, which is then passed to an ontology generator containing the rules. It is also possible to map ontologies through machine learning to transform existing ontologies within the Semantic Web to a format useable in the domain context for the current problem. Doan et al. have introduced the GLUE system to semi-automatically create these semantic mappings using a multi-strategy learning approach based on the joint probability distribution of the compared concepts [19] [20]. The concept is to produce a map between the existing domain and the desired domain that

translates between taxonomies. Future research promises to reduce the human interaction required for ontological engineering.

b) Probabilistic Learning

Elicitation of conditional probabilities to populate distribution tables remains a difficult endeavor, accomplished through SME interview and experimental data collection. Probabilistic learning seeks to reduce the effort involved in establishing prior and conditional probabilities for domain entities by specifying a model using empirical data. Pearl identified two tasks for probabilistic learning [21]:

– Extracting generic hypothesis evidence-relationships from records of experience, and

– Organizing the relationships in a data structure to facilitate recall.

Accuracy and consistency in the PO model could be improved by learning numerical parameters for a given network topology from empirical data instead of relying on SME input. The literature contains numerous techniques for parameter learning; two commonly employed methods are:

Maximum Likelihood [22][23] – Parameters are estimated from a set of empirical data using a likelihood weighting algorithm.

Bayesian Learning [22][23] – Prior knowledge about parameters is encoded and data is treated as evidence to reduce the learning process to calculation of posterior distributions.

Learning is segregated into the categories of structure learning and parameter estimation [23][24]. In parameter estimation, the dependency structure of the probabilistic representation is known. The learning task is to define the parameters of the Local Probability Distributions (LPDs). The goal of structure learning is to extract the structure of the probabilistic representation from the dataset.

Learning a Probabilistic Relational Model (PRM) requires input in the form of a relational schema that describes the set of classes, the attributes associated with the classes, and the relations between objects of classes for the domain. In the parameter estimation task, the structure is given, which defines the parents for each attribute. The parameters that define the Conditional Probability Disributions (CPDs) for the structure are learned using the likelihood function to determine the probability of the dataset given the model. Structure learning of a PRM is more complex and requires a method to find possible structures and then score them. Getoor et al. describes the use of a greedy local search procedure to produce a candidate structure which is then scored using the prior probability of the structure and the probability of the dataset, given the structure [23].

Recall that the structure of a Markov Logic Network (MLN) includes a node for each variable and a potential function for each set of nodes that is pairwise linked. Parameter estimation for MLN is performed by computation of the Markov network weights that represent the clique potential using an optimization of the likelihood function. Structure learning is performed by a greedy algorithm on the network features [25].

Multi-Entity Bayesian Network (MEBN) learning also takes advantage of the structure associated with a relational database. A key component is generation of a MEBN-RM model that specifies a mapping of MEBN elements to the relational model of the database. MEBN parameter learning estimates the parameters of the local distribution for a resident node of an MTheory, given the structure and the database using maximum likelihood estimation. MEBN structure learning organizes random variables into MFrags and identifies parent-child relationships between nodes, given the database. Any Bayesian Network Structure search algorithm may be used [26]. More recently, Park et al. has extended the MEBN learning algorithm to include both discrete and continuous random variables [27].

6) Knowledge Base

The knowledge base is a historic collection of domain-specific knowledge contributed by domain SMEs and may include ontological information (classes, properties, characteristics, and relationships), logical constraints, heuristics, and probabilities. The breadth of knowledge stored within is unspecified. To distinguish the KB from evidence, there is no temporal component associated with the knowledge base; information contained therein may not represent the current domain state. Marakas differentiates a database from a knowledge base in this fashion:

"… a collection of data representing facts is a database. The collection of an expert's set of facts and heuristics is a knowledge base [28]*."*

7) Ontology Structures

Ontologies, including probabilistic ontologies, provide a means to represent knowledge and relationships between hierarchically organized classes of objects. Ontologies exist to enable knowledge sharing and reuse [11] [13]. As a set of definitions of formal vocabulary, ontologies allow knowledge sharing among hierarchically organized entities. A probabilistic ontology addresses the inherent uncertainty involved in inferential reasoning applications with inconclusive evidence by representing it probabilistically.

a) Ontology

A working ontology captures the classes, properties, and the relationships of a domain of interest. Production of this relational framework facilitates comprehension of the hierarchical organization of domain entities; the relationships between and properties of domain entities; as well as causal relationships among entities. When uncertainty about aspects of the domain is important to the purpose for which the ontology is being developed, a probabilistic ontology is needed to represent the uncertainty.

b) Probabilistic Ontology

A probabilistic ontology provides a means to represent and reason with uncertainty by integrating the inferential reasoning power of probabilistic languages with the first-order expressivity of ontologies. Few things are certain, and inferring in the presence of uncertainty allows the decision maker to

focus attention on the most relevant data through designed queries.

C. Support Layer

The Support Layer provides the background technology and design strategy necessary to instantiate the conceptualization of a specific probabilistic ontology to satisfy identified requirements. It includes existing ontologies available for reuse or re-engineering, software tools that enable ontology and probabilistic ontology development, mathematical languages that allow representation of entity attributes and their relationships, and databases of existing facts referenced for learning and knowledge base population. The purpose of the Support Layer is to facilitate probabilistic ontology development by identifying technological and semantic features specific to a particular inferential reasoning model. The four Support Layer components are discussed below.

1) Existing Ontologies

Model reuse is a strength of the ontological engineering discipline and effort should be made to research and incorporate existing ontology material into new application areas. This will reduce overall effort and promote commonality among different products. Some suggested ontology repositories are listed below.

2) Modeling Languages

A modeling language is a graphical or textual representation used to express knowledge, information, processes or systems with a consistent set of rules and syntax. In the RAPOD, modeling languages serve three functions:

- System Architecture Representation
- Object Relationship Representation
- Ontology (and Probabilistic Ontology) Representation

A probabilistic ontology is an extension of an ontology which incorporates uncertainty while respecting its relational structure and domain specificity. The output of the RAPOD is a unique instantiated architecture for development of a domain-specific probabilistic ontology to meet an inferential reasoning requirement. The architecture includes models from each of the above representation categories and may be reused for development of new probabilistic ontologies in similar domains. The following sections describe the purpose of these representations.

a) System Architecture Representation

An architecture is a conceptual design that defines the structure and behavior of a system. There are two types of representations commonly employed: traditional and object-oriented, represented here by IDEF0 and UP.

- Icam Definition for Function Modeling (IDEF0) – IDEF0 is a process modeling technique that focuses on the functional model of a system. The model is expressed as a set of diagrams, often called pages. IDEF0 has been applied to the development of information systems, business processes and hardware systems [5].
- Unified Process (UP) – UP is an iterative, comprehensive development approach adapted to object oriented models, tools and techniques [29]. It was developed initially for software systems, but in recent years has been adapted to systems that include hardware and business processes.

IDEF0 is commonly associated with hardware systems and systems-of-systems, especially within the Department of Defense Architecture Framework (DODAF). Class hierarchies are fundamental to ontologies, and object oriented design is focused on modeling class hierarchies.

b) Object Relationship Representation

Object modeling languages are used to represent relationships at the system and object level of abstraction to enable clear, concise communication between Stakeholder Decision Maker and the PO Developer. While the specific choice of language is often left to the developer, object relationships are frequently represented using languages such as:

- Unified Modeling Language (UML) – UML is a graphical modeling language for the creation of object-oriented models used primarily for software engineering [29].
- Systems Modeling Language (SysML) – SysML extends UML language with semantic foundation for representing requirements, behavior, structure, and properties of systems and components [30] [31].

There are many diagrams and representations appropriate to systems architecting available in both UML and SysML; the PO Developer should select and implement these tools to maximize clear communications with the Stakeholder Decision Maker.

c) Ontology Representation

Ontology languages allow developers to create explicit, formal conceptualizations of domain models. The main requirements of an ontology language identified by Antoniou and Harmelen include [32]:

- Well-defined syntax
- Well-defined semantics
- Efficient reasoning support
- Sufficient expressive power
- Convenience of expression

Ontology languages are formal, declarative representations that allow compilation and organization of knowledge about a domain in formal knowledge structures with clearly defined semantics. Further, they include reasoning rules to represent relationships between knowledge classes. The literature contains many different ontology languages, some of which are optimized for specific domains. Some of the more common examples include [10]:

- Web Ontology Language (OWL) – Created by W3C, derived from DAML+OIL and builds on RDF(S).

- Resource Description Framework (RDF) – Created by W3C as a semantic network based language to describe web resources.
- Knowledge Interchange Format (KIF) (including OntoLingua) – Based on FOL with an underlying frame paradigm, overlaid by OntoLingua to simplify operator functionality.
- DARPA Agent Markup Language + Ontology Inference Layer (DAML+OIL) – Created by US and EU committee, an extension of RDF(S) with datatypes and nominals. DAML+OIL has been superseded by OWL.
- CycL – A declarative language used to represent the knowledge stored in the Cyc Knowledge Base [33].
- Common Logic (CL) – A FOL language for knowledge interchange approved and published as an ISO standard for representation and interchange of information and data among disparate computer systems [34].
- Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) – A FOL reference module of the Wonderweb Project adopted as a starting point for comparing and elucidating relationships between ontologies [35].
- Basic Formal Ontology (BFO) – An upper-level ontological framework used in support of domain ontologies developed for scientific research [36].

OWL has been selected by the World Wide Web Consortium (W3C) as the language of the Semantic Web and has therefore received broad attention in the research and development communities. Further, OWL is the ontology language used by the UnBBayes software tool, allowing evolution of an ontology to a probabilistic ontology without the need to recreate the classes, instances, and relationships in a new tool. Recall that PR-OWL expresses MEBN in OWL [13]. Of the above ontology languages, only OWL allows expression of probabilistic information along with an ontology through the PR-OWL extension.

d) Probabilistic Ontology Representation

Probabilistic ontologies are used to comprehensively describe knowledge about a domain and the uncertainty embedded in that knowledge in a principled, structured and sharable way [13]. The probabilistic web ontology language (PR-OWL) and its successor (PR-OWL 2) provide a knowledge representation formalism with MEBN as the underlying semantics. A MEBN represents knowledge about attributes of entities and their relationships as a collection of similar hypotheses organized into theories which satisfy consistency constraints ensuring a unique joint probability distribution over the random variables of interest [37]. A modeling language is a graphical or textual representation used to express knowledge, information, processes or systems with a consistent set of rules and syntax. In the RAPOD, modeling languages serve three functions:

- System Architecture Representation
- Object Relationship Representation
- Ontology (and Probabilistic Ontology) Representation

A probabilistic ontology is an extension of an ontology which incorporates uncertainty while respecting its relational structure and domain specificity. The output of the RAPOD is a unique instantiated architecture for development of a domain-specific probabilistic ontology to meet an inferential reasoning requirement. The architecture includes models from each of the above representation categories and may be reused for development of new probabilistic ontologies in similar domains. The following sections describe the purpose of these representations.

3) Software Tools

Modeling tools represent the software implementation packages used for development and implementation of architectures, ontologies, and probabilistic ontologies in the chosen modeling language. With the appropriate modeling tools, the entire ontology life cycle may be managed, including design, implementation, enhancement, and support.

A number of tools are available to capture data and model the components of a probabilistic ontology. The PO Developer selects software tools with the correct fidelity to represent relevant viewpoints and provide the desired communication and inferential reasoning representation. A combination of these tools gives the PO Developer flexibility in creating necessary views for communication, as well as operational ontology and probabilistic ontology models.

a) General Purpose Modeling Tools

Creation of a probabilistic ontology requires representation of many abstractions of data, processes, and relationships, each of which may be best represented in a different software application. However, to the extent possible, a single, general-purpose tool should be maximized to enhance readability and consistency. Tools such as Microsoft Visio and MagicDraw assist in visual representation to simplify complex concepts.

b) Ontology Engineering Software Tools

Ontological engineering tools capture the classes, properties, and instances of ontology entities in a hierarchical structure. Further, they describe their relationships, domains and ranges in a contextual environment. The most popular ontological engineering tool is Protégé, currently in version 4.1.0 (build 239). Protégé also has the advantage of integration with UnBBayes, which allows seamless implementation of uncertainty to establish the probabilistic ontology.

c) Probabilistic Ontology Engineering Software Tools

Few tools are able to model the complex integration of probability and ontologies. The most advanced is UnBBayes, an open source product developed by University of Brasilia and enhanced in collaboration with George Mason University. UnBBayes has a PR-OWL plug-in that ingests a Protégé ontology and allows the developer to represent uncertainty within its hierarchical structure through MEBN Fragments using the Probabilistic Web Ontology Language (PR-OWL 2).

III. SUMMARY

Use of a reference architecture facilitates design, implementation, and reuse of a domain-specific probabilistic ontology construction process by specifying the logical choices of components to create a blueprint for a contextual solution. The instantiated architecture is available for reuse to solve like problems in similar domains.

REFERENCES

[1] Office of the Assistance Secretary of Defense for Networks and Information Integration (OASD/NII), "Reference Architecture Description," Arlington, 2010.

[2] Heather Kreger, Vince Brunssen, Robert Sawyer, Ali Arsanjani, and Rob High. (2012, Jan) IBM Developer Works. [Online]. http://www.ibm.com/developerworks/webservices/library/ws-soa-ref-arch/.

[3] Richard J. Haberlin, *Probabilistic Ontology Reference Architecture and Design Methodology*, PhD George Mason University, 2013.

[4] Philippe Kruchten, *The Rational Unified Process: An Introduction*. Upper Saddle River: Addison-Wesley, 2004.

[5] Dennis M. Buede, *The Engineering Design of Systems: Models and Methods*. New York: John Wiley & Sons, 2000.

[6] James E. Armstrong, "Issue Formulation," in *Handbook of Systems Engineering and Management*. Hoboken: John Wiley & Sons, 2009, pp. 1027-1089.

[7] Gerald Kotonya and Ian Sommerville, *Requirements Engineering Processes and Techniques*. Chichester: John Wiley & Sons, 1998.

[8] Jeffrey O. Grady, *System Requirements Analysis*. New York: McGraw-Hill, Inc., 1993.

[9] John M. Green, "Establishing System Measures of Effectiveness," in *Proceedings of the 2nd Biennial National Forum on Weapon System Effectiveness*, Laurel, 2001, pp. 1-5.

[10] Asuncion Gomez-Perez, Fernandez-Lopez Mariano, and Oscar Corcho, *Ontological Engineering with Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web*. London: Springer-Verlag, 2010.

[11] Thomas R. Gruber, "Toward Principles for the Design of Ontologies Used for Knowledge Sharing," *International Journal of Human-Computer Studies*, pp. 907-928, 1995.

[12] Michael K. Bergman, "A Brief Survey of Ontology Development Methodologies," 2011, [Online]. http://www.mkbergman.com/906/a-brief-survey-of-ontology-development-methodologies/

[13] Paulo Cesar G. da Costa. *Bayesian Semantics for the Semantic Web*, PhD George Mason Univeristy, 2005. [Online]. http://hdl.handle.net/1920/455 .

[14] Maria C. Keet, "Dependencies between Ontology Design Parameters," *International Journal of Metadata, Semantics and Ontologies*, pp. 265-284, 2010.

[15] Paul Buitelaar and Bernardo Magnini, "Ontology Learning from Text: An Overview," in *Ontology Learning from Text: Methods, Applications and Evaluation*.: IOS Press, 2005, pp. 3-12.

[16] John Sowa. (2001) Ontology. [Online]. http://www.jfsowa.com/ontology/ .

[17] Khalid Albarrak, *An Extensible Framework for Generating Ontology from Various Data Models*, May 2013, PhD Dissertation.

[18] Man Li, Xiao-Yong Du, and Shan Wang, "Learning Ontology from Relational Database," in *Proceedings of the 4th International Conference on Machine Learning and Cybernetics*, Guangzhou, 2005, pp. 3410-3415.

[19] AnHai Doan, Jayant Madhavan, Pedro Domingos, and Alon Halevy, "Ontology Matching: A Machine Learning Approach," in *Handbook on Ontologies*. Berlin: Springer-Verlag, 2009, pp. 385-404.

[20] Anhai Doan, Jayant Madhavan, Pedro Domingos, and Alon Halevy, "Ontology Matching: A Machine Learning Approach," in *Handbook on Ontologies in Information Systems*.: Springer, 2003, pp. 397-416.

[21] Judea Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco: Morgan Kaufmann, 1988.

[22] Adnan Darwiche, *Modeling and Reasoning with Bayesian Networks*. Cambridge: Cambridge Univeristy Press, 2009.

[23] Lise Getoor, Nir Friedman, Daphne Koller, Avi Pfeffer, and Ben Taskar, "Probabilistic Relational Models," in *Introduction to Statistical Relational Learning*. Cambridge: The MIT Press, 2007, pp. 129-174.

[24] James Cussens, "Logic-based Formalisms for Statistical Relational Learning," in *Introduction to Statistical Relational Learning*. Cambridge: MIT Press, 2007, ch. 9, pp. 269-290.

[25] Pedro Domingos and Matthew Richardson, "Markov Logic: A Unifying Framework for Statistical Relational Learning," in *Introduction to Statistical Relational Learning*. Cambridge: The MIT Press, 2007, pp. 339-371.

[26] Cheol Young Park, Kathryn B. Laskey, Paulo C.G. Costa, and Shou Matsumoto, "Multi-Entity Bayesian Networks Learning for Hybrid Variables in Situation Awareness," in *Proceedings of the 16th International Conference on Information Fusion (submitted)*, Istanbul, 2013, pp. 1-8.

[27] Cheol Young Park, Kathryn B. Laskey, Paulo C.G.N. Costa, and Shou Matsumoto, "Multi-Entity Bayesian Networks Learning in Predictive Situation Awareness," in *Proceedings of the 18th International Command and Control Research and Technology Symposium*, Alexandria, 2013, pp. 1-19.

[28] George M. Marakas, *Decision Support Systems in the 21st Century*. Upper Saddle River: Prentice Hall, 2003.

[29] John W. Satzinger, Robert B. Jackson, and Stephen D. Burd, *Systems Analysis and Design in a Changing World*. Boston: Course Technology, 2004.

[30] Sanford Friedenthal, Alan Moore, and Rick Steiner, *A Practical Guide to SysML: The Systems Modeling Language*. Amsterdam: Elsevier, 2008.

[31] Sanford Friedenthal, Alan Moore, and Rick Steiner, *OMG Systems Modeling Language Tutorial*.: Object Management Group, 2008.

[32] Grigoris Antoniou and Frank Van Harmelen, "Web Ontology Language: OWL," in *Handbook on Ontologies in Information Systems*.: Springer-Verlag, 2003.

[33] Cycorp. (2013, June) CycL: The Cyc Knowledge Representation Language. [Online]. http://www.cyc.com/cyc/cycl .

[34] International Standards Organization, "Information technology - Common Logic (CL): a framework for a family of logic-based languages," International Standards Organization, Standard ISO/IEC 24707:2007(E), 2007.

[35] Institute of Cognitive Science and Technology Italian National Research Council. (2013, June) WonderWeb. [Online]. http://www.loa.istc.cnr.it/DOLCE.html .

[36] Institute for Formal Ontology and Medical Information Science. (2013, March) BFO: Basic Formal Ontology. [Online]. http://www.ifomis.org/bfo .

[37] Paulo Cesar G. da Costa, K.C. Chang, Kathryn B. Laskey, and Rommel Novaes Carvalho, "A Multidisciplinary Approach to High Level Fusion in Predictive Situational Awareness," in *Proceedings of the 11th International Conference of the Society of Information Fusion*, Seattle, 2009.

Focused Belief Measures for Uncertainty Quantification in High Performance Semantic Analysis

Cliff Joslyn
National Security Directorate
Pacific Northwest National Laboratory
Seattle, WA 98109
Email: cliff.joslyn@pnnl.gov

Jesse Weaver
Fundamental and Computational Sciences Directorate
Pacific Northwest National Laboratory
Richland, WA 99354
Email: jesse.weaver@pnnl.gov

***Abstract*—In web-scale semantic data analytics there is a great need for methods which aggregate uncertainty claims, on the one hand respecting the information provided as accurately as possible, while on the other still being tractable. Traditional statistical methods are more robust, but only represent distributional, additive uncertainty. Generalized information theory methods, including fuzzy systems and Dempster-Shafer (DS) evidence theory, represent multiple forms of uncertainty, but are computationally and methodologically difficult. We require methods which provide an effective balance between the complete representation of the full complexity of uncertainty claims in their interaction, while satisfying the needs of both computational complexity and human cognition. Here we build on Jøsang's subjective logic to posit methods in focused belief measures (FBMs), where a full DS structure is focused to a single event. The resulting *ternary* logical structure is posited to be able to capture the *minimally sufficient* amount of generalized complexity needed at a *maximum* of computational efficiency. We demonstrate the efficacy of this approach in a web ingest experiment over the 2012 Billion Triple dataset from the Semantic Web Challenge.**

I. Introduction

Many analytic domains face the problem of determining the veracity of claims from multiple sources. The problem can be further complicated by the presence of large numbers of sources asserting large numbers of propositions over short periods of time. Examples include intelligence gathering and sensor networks. Such problems are only exacerbated on the web with the constituent heterogeneity of data, and then again especially when brought to web scale.

While there are various logics for dealing with inconsistency or uncertainty [3], [13], [14], to our knowledge, none have achieved significant uptake in computational systems for large data. Traditional statistical uncertainty representation (UR) models fail to represent complex uncertainty situations requiring imprecision or other forms of ambiguous judgements. So-called Generalized Information Theory (GIT) approaches [12] such as fuzzy and Dempster-Shafer (DS) can represent these complexities, but at the cost of high computational expense. Massive streaming or ingest problems in the semantic web require UR strategies which provide both representation of ambiguity and computational efficiency.

Here we present Focused Belief Measures (FBMs), an adaptation of Jøsang's subjective logic (SL) [10], as a candidate to play this role. By modifying DS to focus on specific events in a complex space, FBMs can model logical combinations of complex beliefs involving imprecision, ambiguity, or total ignorance using *linear* algorithms. This compromise promises to support the *minimal* amount of generalized complexity which may be nonetheless sufficient, but at a *maximum* of computational efficiency.

We begin by introducing FBMs in the context of both SL and DS. We then demonstrate its utility in a web analytics experiment involving the evaluation of a large RDF graph drawn from the 2012 Semantic Web Challenge.

II. Focused Belief Measures (FBMs)

UR methods and formalisms are legion, primarily rooted in probability theory, logic, or their combination (e.g. [5]). For decades, UR researchers have struggled with two competing imperatives. On the one hand, traditional UR methods, including probabilistic (statistical) and logical approaches require closed-world assumptions. For probability, we require knowledge about likelihood distributions over a set of entities which are both exhaustive and mutually exclusive in order to guarantee mathematical additivity: summation of all modeled probabilities to 1. Representing total uncertainty requires assuming a uniform distribution over these choices. Similarly, traditional logic represents only the two states for true (A) and false ($\sim A$, here taking $\sim$ for negation), according to an excluded middle axiom.

The large range of GIT methods, including fuzzy systems, DS evidence theory, random sets, imprecise probabilities [16], and many others, support open world situations by allowing some form of *third option* or *remainder* between True and False, or non-additive, imprecise "overlap" between non-disjoint options. They do this in different ways, but the basic concept is the same, to generalize traditional approaches by relaxing certain axioms, such as allowing probabilities to sum to more or less than 1, or to recognize truth values "between" True and False (different kinds of "Maybe"), in

order to represent more complex uncertainty structures other than probabilistic likelihoods. In this way one can represent inherent vagueness or imprecision of events, or second- or higher-order uncertainties about other uncertainties, reflecting the veracity of the information source, the confidence or likelihood of claims, the uncertainty about a claim, or other open world situations where "something else" we didn't think about could occur.

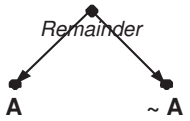


Fig. 1. In this simplest case, generalized information theories support imprecise choices by always including a "third option" or "remainder", allowing positive weight to "neither A nor $\sim A$.

These generalized mathematical structures are inherently hierarchical, since this "remainder" "stands above" its constituent choices. Fig. 1 shows the absolute simplest such case, where "neither A nor $\sim A$" is our "third choice" standing above A and $\sim A$ themselves. This remainder can be used to represent *total* uncertainty or imprecision when positive weight is given to the remainder, but no weight is given to either of the two specific choices.

Fig. 1 actually shows a tiny lattice structure, and the resulting methods require lattice-based computations arising from non-additivity. For example, classical probability has the condition $\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B)$, which is a fully modular function on the subset lattice, allowing relatively simple calculations. But in non-additive formalisms, we have $\Pr(A \cup B) \gtrless \Pr(A) + \Pr(B) - \Pr(A \cap B)$, which is sub- or super-modularity [6]. Modularity allows probabilities of "bigger" events to be calculated from those of "smaller", so non-modularity forces a high computational price. In big data, semantic web environments with massive ingest and streaming input applications, we need methods for representing such hybrid uncertainty, but which are both expressive and tractable.

Our FBM approach is built on Jøsang's SL [10], which is in turn based on DS. Consider a decision problem, like whether Alice, Bob, or Carol committed a crime. In probability, we need $P = p(A) + p(B) + p(C) = 1$ to hold. If $P > 1$, then we have conflict and have to renormalize; while if $P < 1$, then we have a *remainder*, represented as an uncertainty $U = 1 - P > 0$ leftover. In DS theory, we represent general probabilistic uncertainty by giving probabilities m not just to each of these $n = 3$ disjoint events $A, B, C \in \Omega$, but to each of the 2^n *subsets* $R \subseteq \Omega$ of such events. Formally, we have

$$m\colon 2^\Omega \to [0,1], \quad m(\emptyset) = 0, \quad \sum_{R \subseteq \Omega} m(R) = 1.$$

We identify any $R \subseteq \Omega$ with $m(R) > 0$ as *focal*.

This supports modeling of imprecision and ignorance together with likelihood by assigning values to the completely imprecise event $m(\Omega)$, down to the most precise singletons $m(\{A\})$, and everything in between, including *composite* events like $m(\{A, B\})$ for "Alice and/or Bob did it".

The resultant **belief measures**

$$b\colon 2^\Omega \to [0,1], \quad b(R) = \sum_{S \subseteq R} b(S)$$

on any subset $R \subseteq \Omega$ capture a mixture of likelihood and imprecision, since claims m about subsets R cannot necessarily be disambiguated to knowledge about their constituent elements $\omega \in R$. But considering (the middle of) Fig. 2 compared to Fig. 1, we see that we now need to support the full Boolean lattice representing the power set 2^Ω of all subsets. This comes at a huge computational cost, since we now have to work with 2^n rather than n claims, and moreover their interaction within the lattice structure.

But Jøsang [10] has noted that if we *focus* attention to a *particular* composite event $R \subseteq \Omega$ (like $\{A, B\}$=Alice and/or Bob did it), we can reduce the complexity to just three *disjoint* groups of subsets:

1) Those (like $\{A\}$=Alice did it) completely within R supporting R itself;
2) Those (like $\{C\}$=Carol did it) completely disjoint from R supporting $\sim R$ (now taking $\sim R = \Omega \setminus R$ for set complement); and
3) The remainder (like $\{B, C\}$=Bob and/or Carol did it), providing information contradictory to or ambiguous with respect to *both* R and $\sim R$.

These three groups reduce to a single "opinion" vector

$$w(R) = \langle b(R), d(R), u(R) \rangle ,$$

where in addition to $b(R)$ as the **belief** of R, we have

$$d(R) = b(\sim R) = \sum_{S \subseteq \sim R} m(S)$$

as the belief of $\sim R$, that is the **disbelief** of R[1]. Since $b(R), d(R) \in [0,1]$ and $b(R) + d(R) \leq 1$, this allows us to define

$$u(R) = \sum_{S\colon \emptyset \neq S \cap R \neq S} m(S) = 1 - b(R) - d(R)$$

to serve elegantly as our generalized remainder, or **uncertainty** of R.

As so specified, $b(R) + d(R) + u(R) = 1$. Thus b, d, and u exhaust all the options concerning R and $\sim R$, but do so while *including* a representation of the "remainder", u, which is about "neither R nor $\sim R$". This reflects the fact that while,

[1]We depart from Jøsang in using this formulation for d, rather than his (equivalent) $d(R) = \sum_{S \cap R \neq \emptyset, S \not\subseteq R} m(S)$. His is both formally more complex and conceptually less cogent, since it does not capture the sense in which b and d represent the overall belief in the set R and its "opposite". Since our choice is to cast both b and d as beliefs, just in the set R and its opposite $\sim R$, and additionally since our formulation does not rely on anything inherently either "subjective" or "logical", we choose to identify our formulation as **focused belief measures** rather than Jøsang's "subjective logic".

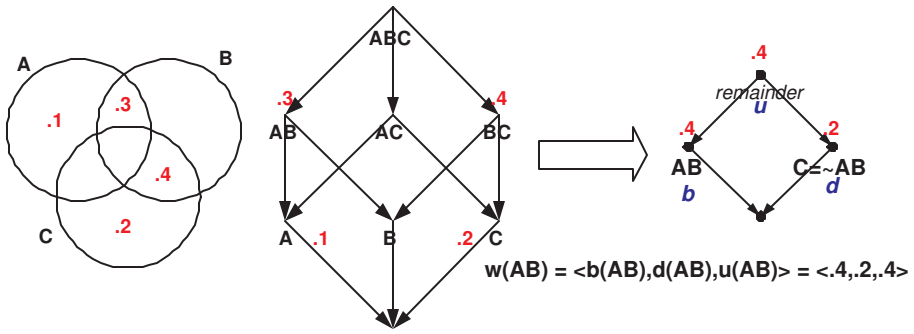


Fig. 2. Proabilities about Alice, Bob, and Carol, *and their combinations*, need $2^3 - 1 = 7$ assessments. When focused on $\{A, B\}$, we reduce to three.

for any $R \subseteq \Omega$, the two *sets* R and $\sim R$ partition Ω, it is rather the three *classes* (sets of subsets)

$$\{S \subseteq R\}, \quad \{S \subseteq \sim R\}, \quad 2^\Omega \setminus (\{S \subseteq R\} \cup \{S \subseteq \sim R\})$$

which partition the power set 2^Ω. It is this third class which is our remainder.

Consider opinions $w_A(R)$, $w_A(S)$, and $w_B(S)$ as opinions from information sources A and B about propositions R and S. Also let $w_A(B)$ be source A's opinion of source B. Jøsang then provides a series of algebraic operators for different combinations, including:

- **Conjunction**

$$w_A(R \wedge S) = \Big\langle b_A(R)b_A(S), \\ d_A(R) + d_A(R) - d_A(R)d_A(S), \\ b_A(R)u_A(R) + u_A(R)b_A(S) + \\ u_A(R)u_A(S) \Big\rangle$$

  and **disjunction**

$$w_A(R \vee S) = \Big\langle b_A(R) + b_A(R) - b_A(R)b_A(S), \\ d_A(R)d_A(S), \\ d_A(R)u_A(R) + u_A(R)d_A(S) + \\ u_A(R)u_A(S) \Big\rangle$$

  opinions about two different propositions by the same source;
- A parallel **consensus** operator

$$w_A(R) \oplus w_B(R) = \left\langle \frac{b_A u_B + b_B u_A}{u_A + u_B - u_A u_B}, \right. \\ \frac{d_A u_B + d_B u_A}{u_A + u_B - u_A u_B}, \\ \left. \frac{u_A u_B}{u_A + u_B - u_A u_B} \right\rangle$$

  expressing the opinion of one proposition R by two sources A, B; and
- A series **discounting** operator

$$w_A(B) \otimes w_B(S) = \Big\langle b_A(B)b_B(S), \\ b_A(B)d_B(S), \\ d_A(B) + u_A(B) + b_A(B)u_B(S) \Big\rangle$$

  expressing the discounted opinion about a base opinion $w_B(S)$ in light of another opinion $w_A(B)$, which we take to be A's opinion about the agent B *expressing* the opinion $w_B(S)$.

Note the tradeoff that FBMs make. We are not representing the full complexity of all $2^n - 1$ possible combinations required by DS; but for any R, or collection of Rs, we are able to directly model $R, \sim R$, *and their remainder*, and while in a *minimal* way, with a *maximal* amount of computational efficiency: the huge advantage of these operators is that they are *linear* in the components b, d, u of the opinion vectors w. Given that we care about only k such events, then in realistic cases we have reduced the size of our problem space to $2k \ll 2^n - 2$ (that is, to $O(k)$ from order $O(2^n)$). We have also vastly improved user comprehensibility, since conceptualizing operations on linear vectors is far less challenging than the structure of hypercubic Boolean lattices. Thus logical combinations of complex situations can be represented easily and cheaply, while still representing our "third option".

This is shown even more strongly in Fig. 3, now the case for $n = 4$ basic choices, shown in the 4-dimensional hypercube (Boolean 4-lattice), and displaying the 4 focal sets. If we wished to track all $2^n - 1 = 15$ possible choices, then so be it, but consider instead that we were only interested in the $k = 3$ choices $\{A\}, \{A, C\}$, and $\{A, B, C\}$. Then we would only need to track the $3k = 9$ pieces of information in the opinion vectors

$$w(A) = \langle .1, .4, .5 \rangle, w(AC) = \langle .3, .4, .3 \rangle$$

$$w(ABC) = \langle .6, 0, .4 \rangle$$

(note that here we simplify notation so that e.g. $ABC = \{A, B, C\}$). As shown, we've replaced the need to store and compute on a 4-dimensional hypercube in exchange for three 2-dimensional hypercubes. FBMs are thus *ternary*, avoiding

limited binary reasoning with a third category to represent "complete ignorance", but also avoiding the full 2^n complexity of the set-based DS.

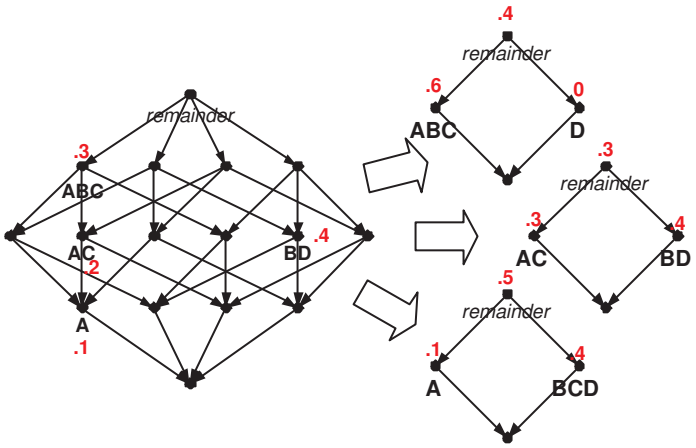


Fig. 3. (Left) Example focal structure for $n = 4$. (Right) Opinion structures for the three target sets $R = A, AC, ABC$.

III. Data Ingest Experiment

We next seek to demonstrate the value, feasability, and tractability of using FBMs in a data ingest experiment on the semantic web. The experiment reported on below is provisional, an initial foray into the basic operation of the FBM approach, but as specifically applied to web-based analytics. In particular, as will be described in more detail below, we use opinion vectors which are relatively constrained examples, being unequivocal for basic claims, but always with residual uncertainty in their aggregation. Further investigation can open the approach up to range over a wider array of values, seeking performance and sensitivity analyses.

A. FBM Problem Setup

Consider the following decision problem. Three data sources make claims about certain facts. Alice and Cindy assert p, but Bob says $\sim p$. The judge must determine whether he/she believes p is true. The judge generally believes Alice and Bob (Bob more than Alice) and thinks that Cindy is lying.

An FBM setup for this problem is shown in Fig. 4. First, the base claims made by Alice, Bob, and Cindy are unequivocally either true or fale (Jøsang calls these "dogmatic"), and thus lack the uncertainty parameter u. We have

$$w_A(p) = \langle 1, 0, 0 \rangle, \quad w_B(p) = \langle 0, 1, 0 \rangle, \quad w_C(p) = \langle 1, 0, 0 \rangle.$$

Next, the judge is inclined to believe Alice at about 80%, but it's not that her remaining 20% *disbelieves* Alice, but she is rather *uncertain* about Alice, not having further grounds to either believe of disbelieve her. We thus have $w(A) = \langle .8, 0, .2 \rangle$. The judge finds Bob convincing at 95%, even more than Alice, so $w(B) = \langle .95, 0, .05 \rangle$. Finally the judge is quite sure that Cindy is lying, but there is always the residual possibility otherwise, so $w(C) = \langle 0, .99, .01 \rangle$.

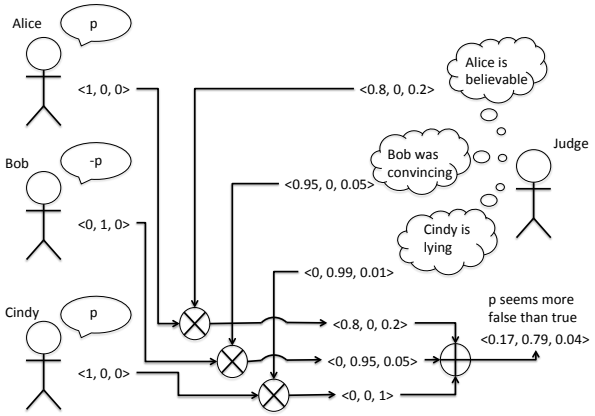


Fig. 4. An FBM problem setup: how to aggregate streaming opinions under source uncertainty?

Assume that Alice, Bob, and Cindy actually testify in order, modeling a streaming ingest operation. Initially, we have absolutely no knowledge about p, and thus the only valid choice is the totally uncertain opinion $w(p) = \langle 0, 0, 1 \rangle$. As an opinion $w_X(p)$ of p by source X arrives, we then update our opinion as:

$$w'(p) = w(p) \oplus (w(X) \otimes w_X(p)),$$

first discounting X's opinion of p with our opinion of X, and then aggregating with our prior opinion. We are thus building the consensus opinions as more data arrives from more sources, and the only state that needs to be saved is a single opinion for p.

After Alice, the judge's opinion of p is

$$\langle 0, 0, 1 \rangle \oplus (\langle 1, 0, 0 \rangle \otimes \langle 0.8, 0, 0.2 \rangle) = \langle 0.8, 0, 0.2 \rangle.$$

Then Bob testifies that p is absolutely false, or alternatively, that $\neg p$ is absolutely true. Now the judge's opinion of p is

$$\langle 0.8, 0, 0.2 \rangle \oplus (\langle 0, 1, 0 \rangle \otimes \langle 0.95, 0, 0.05 \rangle) = \langle 0.17, 0.79, 0.04 \rangle.$$

Bob has effectively changed the judge's mind. Finally, Cindy testifies that p is absolutely true, but the judge is nearly certain that she is lying. In the end, the judge's opinion of p is

$$\langle .17, .79, .04 \rangle \oplus (\langle 1, 0, 0 \rangle \otimes \langle 0, .99, .01 \rangle) = \langle .17, .79, .04 \rangle.$$

The judge's mind did not change at all as a result of Cindy's testimony. In the end, the judge is inclined to believe that p is false.

B. Billion Triple Challenge

Over the last 14 years, the Resource Description Framework (RDF) [2] has become a popular knowledge representation with mature research and tools to support it. It also has associated ontology languages such as RDF Schema (RDFS) [8] and the Web Ontology Language (OWL) [7] which allow

Triples	Number
Any FOAF-related triple	164.3M
With predicate `foaf:name`	7.8M
With predicate `foaf:knows`	17.3M

TABLE I
STATISTICS ABOUT FOAF TRIPLES IN BTC.

for declarative semantics that support reasoning tasks like inference and consistency checking. There are many efforts to expose data as RDF (e.g., Facebook [17], Data.gov [4], biomedical [1], etc.)[2], and major companies are employing RDF to allow users to mark up their data (e.g., Open Graph Protocol[3], schema.org[4], Twitter cards[5]).

Thus there is an abundance of RDF data which is driving the kinds of data integration challenges we posit FBMs to be valuable for. Even those which use different ontologies can be meaningfully unified using a relatively simple "upper ontology" given a basic knowledge of common ontologies [19]. For non-RDF data sources, the heterogeneity problem can be solved by providing an RDF or SPARQL [15] interface on data sources (either on the producer or consumer side, like in [17], which is a coding task) and by providing an appropriate ontology to give a unified view of the data (which is a design task needing sufficiently knowledgeable persons or good documentation of the data source).

We experimented with this streaming FBM method on a large RDF dataset crawled from the web. The 2012 Billion Triple Challenge dataset (BTC)[6] is a set of RDF quads crawled from the Web for the purposes of challenging competitors to work at scale. BTC was chosen because it represents one of the best and largest publicly available RDF datasets, and because of our own past experience working with previous versions of it [11], [18].

The RDF quads in BTC are RDF triples with an additional component that we will refer to as the "graph name". For an RDF quad $\langle s, p, o, g\rangle$ (where g is the graph name), let $d = \text{http}(g)$ be the direct URL of the document retrieved over HTTP when following g (e.g., when you put g in your browser). In some cases, $d = g$. However, http(g) can result in redirection (e.g., HTTP codes 301, 302, 303) in which case $d \neq g$. Many graph names can map to the same document URL, and these mappings are also captured in the (broader sense of the) BTC dataset.

For our experiment, we limited ourselves to quads in BTC that utilized terms from the friends-of-a-friend (FOAF) ontology[7]. FOAF contains 164.3M overall, including two specific sub-groups (`foaf:knows` and `foaf:name`) which we will use below (see Table I).

[2]http://www.semantic-web-journal.net/accepted-datasets – last accessed August 8, 2013 – contains an entire list of such datasets.

[3]http://ogp.me/ – last accessed August 8, 2013

[4]http://schema.org/ – last accessed August 8, 2013

[5]https://dev.twitter.com/docs/cards – last accessed August 8, 2013

[6]http://km.aifb.kit.edu/projects/btc-2012/ – last accessed August 8, 2013

[7]http://xmlns.com/foaf/spec/ – last accessed August 8, 2013

FOAF captures information about people, webpages, and their relationships. We considered documents as sources for determining beliefs. Relating to the judge example, we are the judge, and each document is a witness. We assume every document asserts that it is telling the absolute truth for each triple. That is, for each $\langle s, p, o, g\rangle$ such that $d = \text{http}(g)$, d's belief of $\langle s, p, o\rangle$ is $\langle 1, 0, 0\rangle$ (absolute belief). As the judge, we must determine how to discount these beliefs since we are not inclined to believe anything absolutely just by mere testimony.

Hogan *et al.* [9] have already established a notion of authority for RDF triples based on the sources of the triples, and so we make use of that. It is important to understand that whether an RDF triple is authoritative depends on the relationship between the subject of the RDF triple (that is, s in $\langle s, p, o\rangle$) and the graph name g and/or document URL d. The general idea is that any RDF triple in which the subject is a term defined by the source, such an RDF triple is considered authoritative. The foundation for this notion is laid by concepts of RDF namespaces and Linked Data principles[8], the scope of which are beyond this particular work.

The specific rules we used are as follows.

- For any URI u, let nofrag(u) be everything before the fragment (if any fragment exists). Thus nofrag(data:abc) = data:abc, and nofrag(data:abc#def) = data:abc (the "fragment" is everything after and including the first "#" in a URI). $\langle s, p, o, g\rangle$ is considered authoritative iff nofrag(s)=nofrag(g) or nofrag(s)=nofrag(http(g)).
- We transform $\langle s, p, o, g\rangle$ RDF quads into quints $\langle s, p, o, d, a\rangle$ where $d = \text{http}(g)$ and $a = 1$ if $\langle s, p, o, g\rangle$ is authoritative and $a = 0$ otherwise, and we consider only unique quints. If $a = 1$, our belief in d for the assertion of $\langle s, p, o\rangle$ is $\langle 0.9, 0, 0.1\rangle$ (90% belief), and if $a = 0$, our belief in d for the assertion of $\langle s, p, o\rangle$ is $\langle 0.01, 0, 0.99\rangle$ (99% uncertainty). The values are chosen somewhat arbitrarily. Since d's belief of $\langle s, p, o\rangle$ is always $\langle 1, 0, 0\rangle$ and $X \otimes \langle 1, 0, 0\rangle = X$, our belief in d becomes our belief of d's assertion of $\langle s, p, o\rangle$. (That is, unsurprisingly, our belief in the source for a particular triple is the same as our belief in the triple stated by that source.)
- At this point, we have beliefs for every unique $\langle s, p, o, d, a\rangle$, which we will denote $b(\langle s, p, o, d, a\rangle)$, but we wish to form some overall belief for each unique $\langle s, p, o\rangle$. This is determined using the consensus operator $\oplus$. For every $\langle s, p, o\rangle$, our belief is

$$b(\langle s, p, o\rangle) = \bigoplus_{\langle s,p,o,d,a\rangle \in BTC} b(\langle s, p, o, d, a\rangle).$$

C. Implementation

Our evaluation was run using simple Unix commands like `sort`, `cut`, and `uniq`; and short, custom Perl scripts. This was simply the easiest path to a preliminary evaluation of FBMs. In principle, though, the same computation is easily parallelizable. Let S be a set of data sources (documents),

[8]http://www.w3.org/DesignIssues/LinkedData.html – last accessed August 8, 2013

and let W be a function associating our opinions *about* data sources in S to those same data sources. Let K (the "knowledge base") be a set of opinions *from* the sources in S. Then generalizing the previous formula for BTC, we form our opinion $w(R)$ of a proposition R as:

$$w(R) = \bigoplus_{w_A(R) \in K} [W(A) \otimes w_A(R)].$$

To parallelize this computation, simply artibrarily distribute K to some n processors, that is, $K = \bigcup_{i=0}^{n-1} K_i$. Then each processor can determine its *local* consensus opinions K_i' as:

$$K_i' = \left\{ \bigoplus_{w_A(R) \in K_i} \left[W(A) \otimes w_A(R)\right] \;\middle|\; \forall R \right\}.$$

Then in order to derive the *global* consensus opinions, a simple parallel reduction using the $\oplus$ operator is possible by virtue of the fact that $\oplus$ is associative and commutative. For example, each processor i may have a *local* consensus opinion $w_i(R)$. Then the *global* consensus opinion of R is $\bigoplus_{i=0}^{n-1} w_i(R)$, derived using a parallel reduction. Alternatively, for every proposition R, a hash function h can be used to redistribute *local* consensus opinions among processors so that processor $h(R) \bmod n$ has all of $\{w_i(R)\}_{i=0}^{n-1}$ and the derivation of the *global* opinions can be performed as local operations.

D. Results

The distribution of the consensus beliefs are illustrated in Fig. 5. Each $\langle X, Y \rangle$ point is the number Y of unique $\langle s, p, o \rangle$ triples for which our belief is X. The red points represent authoritative triples, and the blue points represent non-authoritative triples. For a closer view of the highest beliefs, refer to Fig. 6.

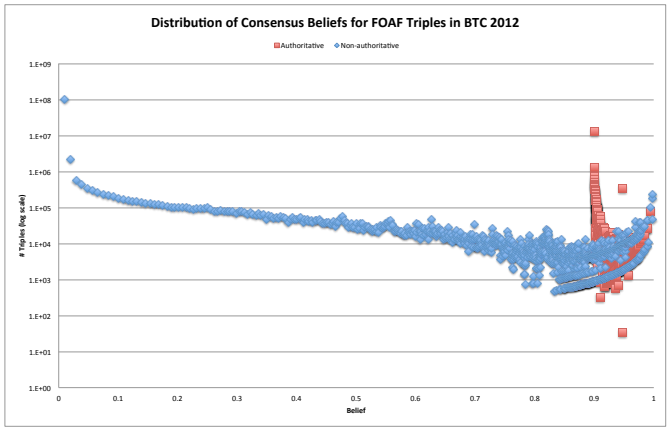


Fig. 5. Each $\langle X, Y \rangle$ point is the number Y of unique $\langle s, p, o \rangle$ triples for which our belief is X. The red points represent authoritative triples, and the blue points represent non-authoritative triples.

We note a good distribution over the range of belief values (i.e., there are no significant gaps across the horizontal axis) which suggests that belief as specified herein provides a useful ranking mechanism. Second, there are a large number of triples

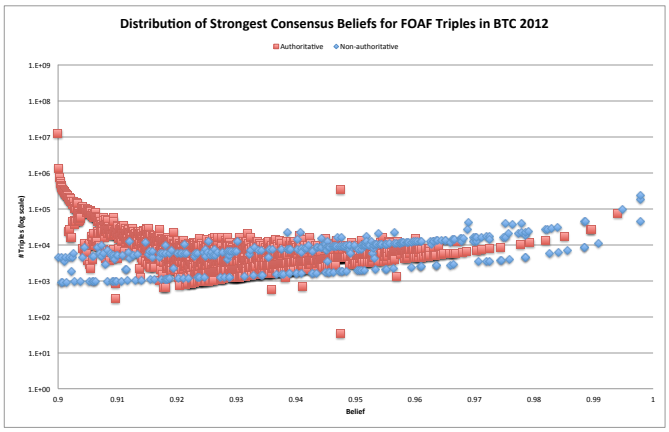


Fig. 6. Each $\langle X, Y \rangle$ point is the number Y of unique $\langle s, p, o \rangle$ triples for which our belief is X. The red points represent authoritative triples, and the blue points represent non-authoritative triples.

for which we have very little belief and a large number of triples for which we have high belief, which indicates that belief as specified herein is a useful metric for separating highly believable statements from hardly believable statements (as long as the underlying assumptions of authority hold and are meaningful in reality). Third, our most believed triples are actually non-authoritative, reflecting strong public consensus about these triples/propositions even without an authoritative source.

Diving deeper into the data, it appears that the overall shape of the charts is caused (at least in part) by distribution of names, as shown in figure 7. It so happens that documents often include the names of people mentioned even if the document is not authoritative for that person. For example, Jesse may have a document that is authoritative for statements about j:jesse, and so the document may say that ⟨j:jesse, foaf:knows, c:cliff⟩ and also that ⟨c:cliff, foaf:name, "Cliff"⟩, even though the document is only authoritative for j:jesse and not c:cliff. We conjecture that popular persons have their names replicated across relatively non-authoritative documents which accounts for the high belief in some non-authoritative triples.

If we look at only triples with foaf:knows as a predicate, the disparity in Fig. 8 is quite obvious. Non-authoritative foaf:knows triples are hardly believed while authoritative foaf:knows triples are very believed. We conjecture that this is because the publication behavior of foaf:knows triples is opposite to that of foaf:name triples. For example, Jesse's document may state that ⟨j:jesse, foaf:knows, c:cliff⟩, but it does not state that ⟨c:cliff, foaf:knows, j:jesse⟩. Clearly, such triples of the latter case exist or else no non-authoritative foaf:knows triples would even exist, but such triples are uncommon which leads to low belief.

IV. Conclusion

This work represents the beginning of our investigation, and indeed more is necessary to verify our conjectures and to

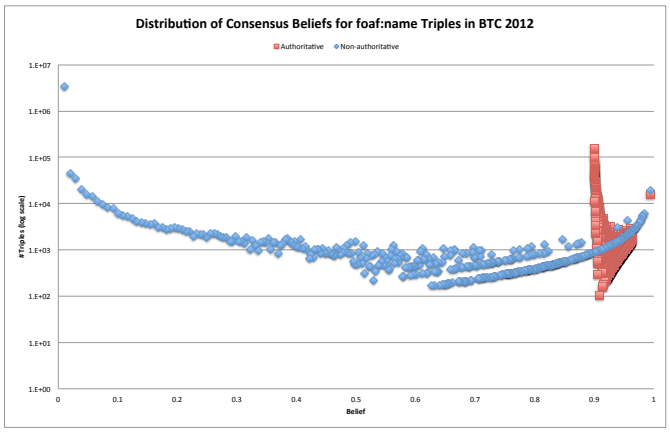


Fig. 7. Belief distribution for the 7.8M triples with predicate foaf:name.

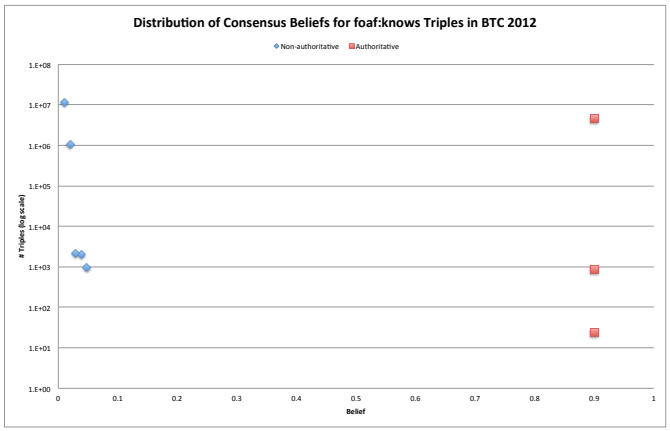


Fig. 8. Belief distribution for the 17.3M triples with predicate foaf:knows in BTC.

find more patterns. Regardless, this preliminary work indicates that focused belief measures hold promise and that more investigation is warranted.

The most significant issue is in our use of "dogmatic" base claims, that is, opinions of the form $\langle 1, 0, 0\rangle$ or $\langle 0, 1, 0\rangle$, expressing complete belief or disbelief on the part of the claimant. In fact, truth claims come in all forms, e.g. "A believes that p" or "A holds p with 50% probability" or "A believes that p falls in the range $[10, 20]$", and many other possible forms involving intervals, distributions, statistical properties, etc. Being able to map these source claims to FBM opinions is an important next problem for us.

Other future work includes:

- The current experiments depend on a number of constant assumptions, as shown above. Parameterization of these constants will support a sensitivity analysis over this space of inputs to help determine experimental behavior.
- Discovering more complex categorizations of triples on the web than merely "authoritative" and "non-authoritative" and determining meaningful discounting opinions for these categorizations.
- Taking into account negative assertions (that is, asserting the falsity of a triple). Such is supported by OWL, but it is expressed using multiple triples. Our evaluation herein equated each single triple as a single proposition.
- Implementation of a parallel system for deriving consensus opinions as described in section III-C.

References

[1] Michael Bada, Kevin Livingston, and Lawrence Hunter. An ontology representation of biomedical data sources and records. *Bio-Ontologies 2011*, 2011.

[2] Jeremy J. Carroll and Graham Klyne. Resource description framework (RDF): Concepts and abstract syntax. W3C recommendation, W3C, February 2004. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

[3] Brian F Chellas. *Modal logic*. Cambridge university press, 1980.

[4] Li Ding, Dominic DiFranzo, Alvaro Graves, James R. Michaelis, Xian Li, Deborah L. McGuinness, and James A. Hendler. TWC data-gov corpus: incrementally generating linked government data from data.gov. In *Proceedings of the 19th International Conference on the World Wide Web*, 2010.

[5] Pedro Domingos and Daniel Lowd. *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan and Claypool, 2009.

[6] Michel Grabisch. Belief functions on lattices. *Int. J. Intelligent Systems*, 24:1:76–95, 2009.

[7] W3C OWL Working Group. OWL 2 web ontology language document overview. Technical report, W3C, December 2012. http://www.w3.org/TR/2012/REC-owl2-overview-20121211/.

[8] Patrick Hayes. RDF semantics. W3C recommendation, W3C, February 2004. http://www.w3.org/TR/2004/REC-rdf-mt-20040210/.

[9] Aidan Hogan, Jeff Z. Pan, Axel Polleres, and Stefan Decker. Saor: template rule optimisations for distributed reasoning over 1 billion linked data triples. In *Proceedings of the 9th international semantic web conference*, pages 337–353, 2010.

[10] Audun Josang. A logic for uncertain probabilities. *Int. J. Uncertainty, Fuzziness and Knowledge-Based Systems*, 9:3:279–311, 2001.

[11] Cliff Joslyn, Bob Adolf, Sinan al Saffar, J Feo, Eric Goodman, David Haglin, Greg Mackey, and David Mizell. High performance descriptive semantic analysis of semantic graph databases. In *Proc. Wshop. on High Performance Computing for the Semantic Web (HPCSW 2011), CEUR*, volume 736, 2011.

[12] Cliff Joslyn and Jane Booker. Generalized information theory for engineering modeling and simulation. In E Nikolaidis et al., editor, *Engineering Design Reliability Handbook*, pages 9:1–40. CRC Press, 2005.

[13] Nils J Nilsson. Probabilistic logic. *Artificial intelligence*, 28(1):71–87, 1986.

[14] Donald Nute. Defeasible logic. In Oskar Bartenstein, Ulrich Geske, Markus Hannebauer, and Osamu Yoshie, editors, *Web Knowledge Management and Decision Support*, volume 2543 of *Lecture Notes in Computer Science*, pages 151–169. Springer Berlin Heidelberg, 2003.

[15] Eric Prud'hommeaux and Andy Seaborne. SPARQL query language for RDF. W3C recommendation, W3C, January 2008. http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/.

[16] P Walley. Towards a unified theory of imprecise probabilities. *Int. J. Approximate Reasoning*, 24:125–148, 2000.

[17] Jesse Weaver and Paul Tarjan. Facebook Linked Data via the Graph API. *Semantic Web Journal*, 2012.

[18] Gregory T Williams, Jesse Weaver, Medha Atre, and James A Hendler. Scalable reduction of large datasets to interesting subsets. In *Billion Triple Challenge, ISWC 2009*, 2009.

[19] Gregory Todd Williams, Jesse Weaver, Medha Atre, and James A Hendler. Scalable reduction of large datasets to interesting subsets. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(4):365–373, 2010.

Recognizing and Countering Biases in Intelligence Analysis with TIACRITIS

Gheorghe Tecuci, David Schum, Dorin Marcu, Mihai Boicu
Learning Agents Center, Volgenau School of Engineering, George Mason University, Fairfax, VA 22030, USA

***Abstract*— This paper discusses different biases which have been identified in Intelligence Analysis and how TIACRITIS, a knowledge-based cognitive assistant for evidence-based hypotheses analysis, can help recognize and partially counter them. After reviewing the architecture of TIACRITIS, the paper shows how it helps recognize and counter many of the analysts' biases in the evaluation of evidence, in the perception of cause and effect, in the estimation of probabilities, and in the retrospective evaluation of intelligence reports. Then the paper introduces three other types of bias that are rarely discussed, biases of the sources of testimonial evidence, biases in the chain of custody of evidence, and biases of the consumers of intelligence, which can also be recognized and countered with TIACRITIS.**

Bias, cognitive assistant, intelligence analysis, evidence-based reasoning, argumentation, symbolic probabilities.

I. INTRODUCTION

Intelligence analysts face the difficult task of drawing defensible and persuasive conclusions from masses of evidence, requiring the development of often stunningly complex arguments that establish and defend the three major credentials of evidence: relevance, believability, and inferential force [1]. This highly complex task is affected by various biases which are inclinations or preferences that interfere with impartial judgment. Some of the biases are due to our simplified information processing strategies that lead to consistent and predictable mental errors. These errors remain compelling even when one is fully aware of their nature, and are therefore exceedingly difficult to overcome [2, p.111-112].

In this paper we propose an approach to the identification and countering of the biases in intelligence analysis. The approach is based on the observation that the best protection against biases comes from the collaborative effort of teams of analysts, who become skilled in the evidential and argumentational elements of their tasks, and who are willing to share their insights with colleagues, who are also willing to listen. As we discuss in this paper, this could be achieved by employing an intelligent analytic tool like TIACRITIS [3] which helps the analyst perform a rigorous evidence-based hypothesis analysis that makes explicit all the reasoning steps, probabilistic assessments, and assumptions, so that they can be critically analyzed and debated. The name TIACRITIS is an abbreviation of Teaching Intelligence Analysts Critical Thinking Skills, which was the initial motivation of developing this system. The system was later extended to also support its use for regular analysis.

In the next section we introduce the architecture of the TIACRITIS cognitive assistant which is based on semantic technologies for knowledge representation, reasoning, and learning. Then, in Section III, we address the analysts' biases discussed by Heuer [2, pp.111-171]: biases in the evaluation of evidence, in the perception of cause and effect, in the estimation of probabilities, and in the retrospective evaluation of intelligence reports. After that we address three other origins of bias that are rarely discussed, even though they may be at least as important on occasion as any analysts' biases.

II. THE TIACRITIS COGNITIVE ASSISTANT

TIACRITIS is a knowledge-based system that supports an intelligence analyst in performing evidence-based hypothesis analysis in the framework of the scientific method. It guides the analyst to view intelligence analysis as ceaseless discovery of *evidence*, *hypotheses*, and *arguments* in a non-stationary world, involving collaborative processes of *evidence in search of hypotheses*, *hypotheses in search of evidence*, and *evidentiary testing of hypotheses* [1, 3]. Fig.1 is an abstract illustration of this astonishingly complex process. First we search for possible hypotheses that would explain a surprising observation **E*** (see the left side of Fig.1): It is possible that **F** might be true. Therefore **G** might be true. Therefore **H**, a hypothesis of high interest, might be true. The problem with drawing this conclusion, however, is that there are other hypotheses that also explain **E***, such as **F'**, **G'**, and **H'**. To conclude **H** we would need to assess all the competing hypotheses, showing that **F**, **G**, and **H** are more likely than their competitors.

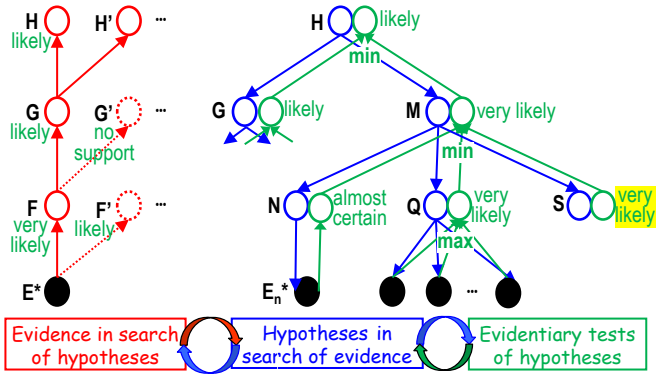


Fig. 1. Scientific method framework of TIACRITIS.

Let us assume that we have shown that **F** and **G** are more likely than their corresponding competing hypotheses. Next we have to assess **H**, **H'**, … . To assess **H** we need additional evidence which is obtained by successively decomposing **H** into simpler and simpler hypotheses, as shown by the blue tree in the right part of Fig.1. **H** would be true if **G** and **M** would be true. Then **M** would be true if **N**, **Q**, and **S** would be true. But if **N** would be true, then we would need to observe evidence $\mathbf{E_n}$*. So we look for $\mathbf{E_n}$* and we may or may not find it. This is the

This research was partially supported by the Department of Defense and by George Mason University. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Department of Defense or the U.S. Government.

process of hypotheses in search of evidence that guides the evidence collection task. Now some of the newly discovered items of evidence (e.g. $\mathbf{E_n}$*) may trigger new hypotheses, or the refinement of the current hypotheses. Therefore, as indicated at the bottom part of Fig.1, the processes of evidence in search of hypotheses and hypotheses in search of evidence take place at the same time, and in response to one another.

Then we use all the collected evidence to assess the hypothesis H. This assessment is probabilistic in nature because the evidence is always *incomplete*, usually *inconclusive*, frequently *ambiguous*, commonly *dissonant*, and has various degrees of *believability* [1]. In the computational theory of intelligence analysis we have developed [3], hypotheses assessment is based on a combination of ideas from the Baconian probability system [4] and the Fuzzy probability system [5], and uses a symbolic probability scale. In particular, in the latest version of TIACRITIS, the likeliness of a hypothesis may have one of the following ordered values:

no support < likely < very likely < almost certain < certain

In this scale, "no support" means that our evidence does not support the conclusion that the hypothesis is true. This may, however, change if new evidence favoring the hypothesis is later discovered. The likeliness of an upper-level hypothesis (e.g., **H**) is obtained from the likeliness of its sub-hypotheses (i.e., **G** and **M**) by using min or max Baconian and Fuzzy combination functions, depending on whether the sub-hypotheses **G** and **M** represent necessary and sufficient conditions for the hypothesis H, sufficient conditions, or just indicators. Competing hypotheses (e.g., **H'**) are assessed in a similar way and the most likely hypothesis is selected. But if no hypothesis is more likely than all its competitors, then the processes of hypotheses in search of evidence, and evidence in search of hypotheses have to be resumed.

TIACRITIS was developed by first customizing the Disciple learning agent shell (a general agent building tool [6, 7]) into a learning agent shell for intelligence analysis, and then by training it with analysis knowledge from several domains [8]. The overall architecture of the Disciple learning agent shell for intelligence analysis is shown in Fig. 2. It contains integrated modules for ontology development, rule learning, problem solving and evidence-based reasoning, mixed-initiative interaction, and tutoring, as well as a hierarchically organized repository of knowledge bases (KB). At the top level of this repository is the general knowledge base for intelligence analysis (IA KB) which contains knowledge applicable to the evidence-based analysis of any type of intelligence hypothesis, from any domain. Under it, and inheriting from it, are domain-specific knowledge bases. Each such Domain KB contains knowledge specific to a particular type of IA problems, such as predictive analysis related to energy sources, or assessments related to the current production of weapons of mass destruction by various actors. Under each Domain KB there are several Scenario KBs, each corresponding to an instance of a problem pattern from that domain, such as, "Assess whether the United States will be a world leader in wind power within the next decade." This particular Scenario KB contains specific knowledge about the United States, as well as items of evidence to make the corresponding analysis. The actual analysis is done by using this knowledge as well as more general knowledge inherited from the corresponding Domain KB and from the IA KB.

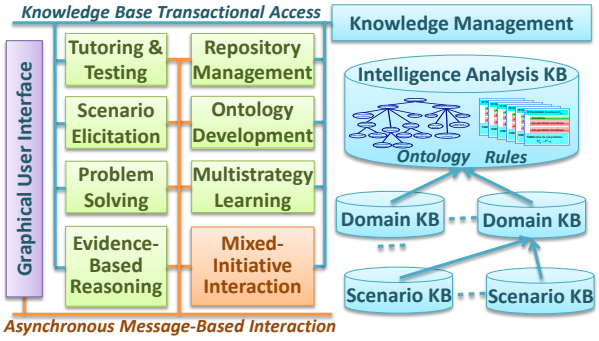


Fig. 2. Learning agent shell for intelligence anlaysis.

Each of these knowledge bases is structured into an ontology of concepts and a set of general problem solving rules expressed with these concepts. The rules are learned from specific examples of reasoning steps, by using the ontology as a generalization hierarchy [7]. The learning agent shell for intelligence analysis was obtained by training the Disciple learning agent shell with general intelligence analysis know-

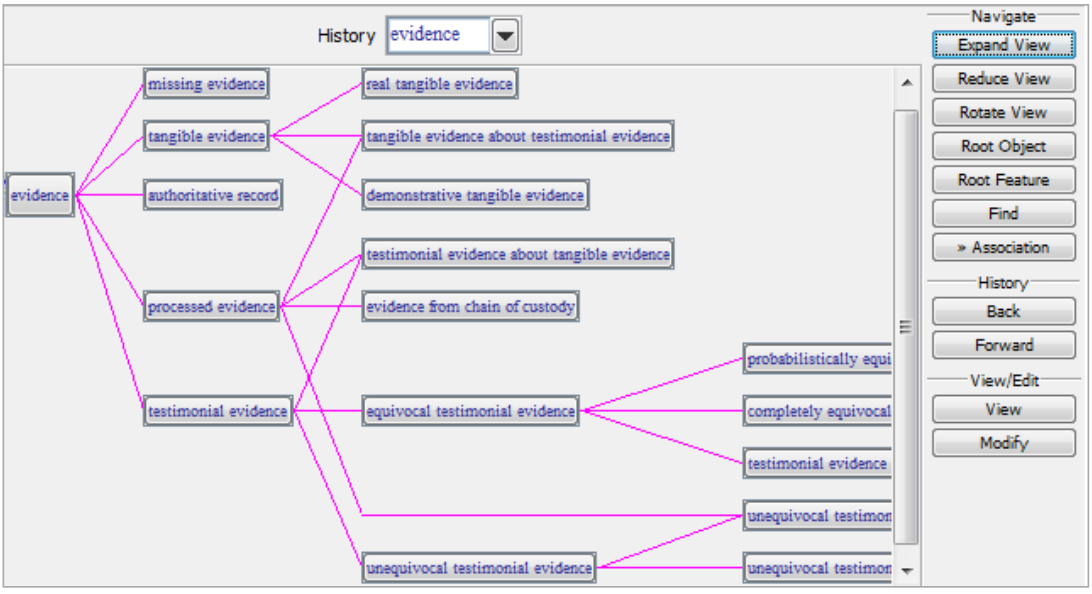


Fig. 3. Ontology fragment showing various types of evidence.

ledge resulting in the development of the IA KB. The IA KB contains both a general ontology and a set of general reasoning rules which are necessary for any Disciple agent for intelligence analysis, as we will briefly present in the following. For example, Fig. 3 shows a general ontology of evidence. It includes both basic types (e.g., testimonial evidence and tangible evidence), as well as evidence mixtures (e.g., testimonial evidence about tangible evidence). The ontology language of Disciple is an extension of RDFS [9] with additional features to facilitate learning [6, 7, 10].

Learned general rules from the IA KB include those for directly assessing a hypothesis based on evidence. These rules automatically reduce the assessment of a leaf hypothesis, such as **Q** in Fig.1, to assessments based on favoring and disfavoring evidence and, further down, to the assessment of the *relevance* and the *believability* of each item of evidence with respect to **Q**. Once these assessments are made, they are combined, from bottom-up, to obtain the *inferential force* of all the items of evidence on **Q**, which results in the likeliness of **Q**.

An example of a learned rule is shown in Fig. 4. It is an if-then problem reduction rule that expresses how and under what conditions a generic hypothesis can be reduced to simpler generic hypotheses. The conditions are represented as first-order logical expressions [7]. In particular, this rule states that, in order to assess the believability of unequivocal testimonial evidence obtained at second hand, one needs to assess both the believability of our source, and the believability of the source of our source. It is by the application of such rules that an agent can generate the reduction part of the trees in Fig.1 and Fig.5.

IF:	Assess the *?O1* of *?O2*.

Q:	What kind of evidence is *?O2*?
A:	unequivocal testimonial evidence obtained at second hand by *?O3* who obtained the information from *?O4*.

MAIN CONDITION

Var	Lower Bound	Upper Bound
?O1	believability	believability
?O2	(unequivocal testimonial evidence obtained at second hand)	(unequivocal testimonial evidence obtained at second hand)
?O3	(person)	(actor)
?O4	(person)	(actor)
?O5	(testimonial evidence)	(testimonial evidence)

Var	Relationship	Var
?O2	has as source	*?O3*
?O2	is obtained from	*?O5*
?O5	has as source	*?O4*

THEN:	Assess the *?O1* of *?O3* as the reporter of *?O2*, an unequivocal testimonial evidence obtained at second hand from *?O4*.
	Assess the *?O1* of *?O4* as the source of *?O2*, an unequivocal testimonial evidence obtained at second hand.

Fig. 4. Learned rule for believability analysis.

The ontology and the rules from the knowledge repository of TIACRITIS allow it to support the analyst in formulating hypotheses, developing arguments that reduce complex hypotheses to simpler and simpler ones (as discussed above), collecting evidence relevant to the simplest hypotheses, and finally assessing the relevance, the believability, and the inferential force of evidence, and the likeliness of the hypotheses. Additionally, TIACRITIS continuously learns from the performed analyses.

As discussed in the rest of this paper, TIACRITIS has one additional important capability. It supports the analysts in recognizing and countering many of their biases. Because Heuer has made a detailed and very well-known analysis of biases in intelligence analysis [2, pp.111-171], we follow his classification and identified characteristic of biases to show how TIACRITIS helps recognizing and countering many of them.

III. BIASES OF THE ANALYST

A. Biases in the Evaluation of Evidence

Heuer first mentions *vividness of evidence* as a necessary criterion for establishing its force. Analysts, like other persons, have preferences for certain kinds of evidence and these preferences can induce biases. In particular, analysts can have a distinct preference for vivid or concrete evidence when less vivid or concrete evidence may be more inferentially valuable. In addition, their personal observations may be over-valued.

First, as discussed in the previous section, the hypothesis in search of evidence phase of the analysis helps identify a wide range of evidentiary needs. For example, the argumentation in Fig. 1 shows that we need evidence relevant to N, evidence relevant to Q, evidence relevant to S, etc. It is unlikely that we would have vivid evidence for each basic hypothesis. So we would be forced to use less vivid evidence as well.

Second, as illustrated by the abstract analysis example in Fig. 5 and discussed in the following, TIACRITIS guides us to assess a simple hypothesis **Q** by performing a uniform, detailed, and systematic evaluation of each item of evidence, *regardless of its "vividness"*, helping us be more objective in the evaluation of the force of evidence.

Let us first consider how to assess the probability of **Q** based only on one item of favoring evidence $\mathbf{E_k}^*$ (see the bottom of Fig. 5). First notice that we call this *likeliness* of **Q**, and not *likelihood*, because in classic probability theory likelihood is $P(\mathbf{E_k}^*|\mathbf{Q})$, while here we are interested in $P(\mathbf{Q}|\mathbf{E_k}^*)$, the posterior probability of **Q** given $\mathbf{E_k}^*$. With TIACRITIS, to assess **Q** *based only on* $\mathbf{E_k}^*$, we have three judgments to make by answering three questions:

The *relevance* question is: *How likely is* ***Q****, based only on* $\boldsymbol{E_k}^*$ *and assuming that* $\boldsymbol{E_k}^*$ *is true?* If $\mathbf{E_k}^*$ favors **Q**, then our answer should be one of the values from "likely" to "certain." If $\mathbf{E_k}^*$ is not relevant to **Q** then our answer should be "no support" because $\mathbf{E_k}^*$ provides no support for the truthfulness of **Q**. If, however, $\mathbf{E_k}^*$ disfavors **Q**, then it favors the negation (or complement) of **Q**, and it should be moved under $\mathbf{Q}^c$.

The *believability* question is: *How likely is it that* $\boldsymbol{E_k}^*$ *is true?* Here the answer should be one of the values from "no

support" to "certain." "Certain" means that we are sure that the event $\mathbf{E_k}$ reported in $\mathbf{E_k}$* did indeed happen. "No support" means that $\mathbf{E_k}$* provides us no reason to believe that the event $\mathbf{E_k}$ reported in $\mathbf{E_k}$* did happen. For example, we believe that the source of $\mathbf{E_k}$* has lied to us.

The *inferential force* question is: *How likely is* ***Q*** *based only on* $\boldsymbol{E_k}$*? TIACRITIS automatically computes this answer as the minimum of the relevance and believability answers. Indeed, to believe that **Q** is true *based only on* $\mathbf{E_k}$*, $\mathbf{E_k}$* should be both relevant to **Q** and believable.

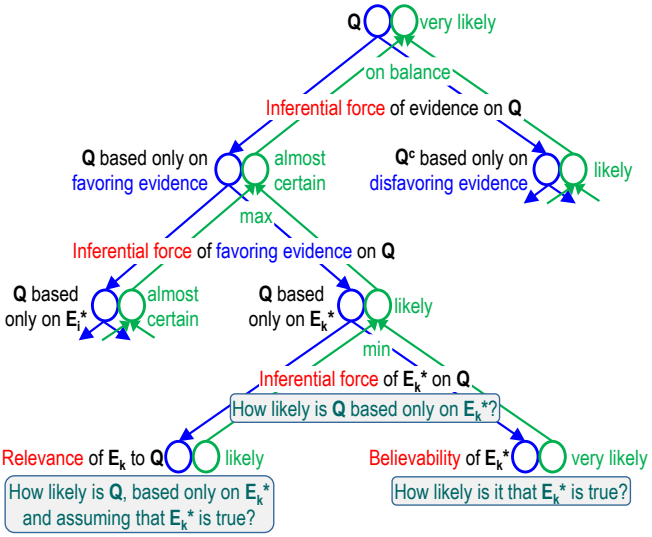


Fig. 5. The relevance, believability, and inferential force of evidence.

When we assess a hypothesis **Q** we may have several items of evidence, some favoring it and some disfavoring it. The favoring evidence is used to assess the likeliness of **Q** and the disfavoring evidence to assess the likeliness of $\mathbf{Q^c}$. Because disfavoring evidence for **Q** is favoring evidence for $\mathbf{Q^c}$, the assessment process for $\mathbf{Q^c}$ is similar to the assessment for **Q**.

When we have several items of favoring evidence, we evaluate **Q** based on each of them (as was explained above), and then we compose the obtained results. This is illustrated in Fig.5 where the assessment of **Q** *based only on* $\mathbf{E_i}$* (almost certain) is composed with the assessment of **Q** *based only on* $\mathbf{E_k}$* (likely), through the maximum function, to obtain the assessment of **Q** *based only on favoring evidence* (almost certain). In this case the use of the maximum function is justified because it is enough to have one item of evidence that is both very relevant and very believable to make us believe that the hypothesis is true.

Let us now assume that $\mathbf{Q^c}$ *based only on disfavoring evidence* is "likely." How should we combine this with the assessment of **Q** *based only on favoring evidence*? As shown at the top of Fig.5, TIACRITIS uses an *on balance* judgment: Because **Q** is "almost certain" and $\mathbf{Q^c}$ is "likely," it concludes that, *based on all available evidence*, **Q** is "very likely."

Heuer also mentions the *absence of evidence* as another origin of bias. The bias here concerns a failure to consider the degree of completeness of available evidence. Consider again the argumentation from Fig. 1 which decomposes complex hypotheses into simpler sub-hypotheses that are assessed based on evidence. This argumentation structure makes very clear that **S** is not supported by any evidence. Thus the analyst should lower her confidence in the final conclusion, countering the *absence of evidence* bias.

The next source of bias mentioned by Heuer is a related one: *oversensitivity to evidence consistency, and not enough concern about the amount of evidence we have*. This kind of bias can easily manifest when using an analytic tool like Heuer's ACH [11] where the analyst judges alternative hypotheses based on evidence, without building any argumentation. With TIACRITIS, the argumentation will reveal if most of the evidence is only relevant to a small fraction of sub-hypotheses, while many other sub-hypotheses have no evidentiary support. For example, the argumentation from Fig. 1 shows that most of the evidence is related to hypothesis **Q**.

According to Heuer [2, pp. 121-122]: "When working with a small but consistent body of evidence, analysts need to consider how representative that evidence is of the total body of potentially available information." The argumentation from Fig. 1 makes very clear that the available evidence is not representative of all the potentially available information. We have no evidence relevant to **S**. If we would later find such evidence which would indicate "no support" for **S**, then the considered argumentation would provide "no support" for the top-level hypothesis **H**. When faced with sub-hypotheses for which there is no evidence (e.g., **S** in Fig. 1), TIACRITIS allows the analyst to consider various what-if scenarios, making alternative assumptions with respect to the likeliness of **S**, and determining their influence on the likeliness of **H**. This should inform the analyst on how to adjust her confidence in the analytic conclusion, to counter the oversensitivity to evidence consistency bias.

Finally, Heuer lists the *persistence of impressions based on discredited evidence* as an origin of bias. If Heuer had written his book in 2003, he might have used the case of Curveball as a very good example [12]. In this case, Curveball's evidence was discredited on a number of grounds but was still believed and taken seriously by some analysts as well as many others.

TIACRITIS helps countering this bias by incorporating in the argumentation an explicit analysis of the believability of evidence, especially for key evidence that has a direct influence on the analytic conclusion. When such an evidence item is discredited, specific elements of its analysis are updated, and this leads to the automatic updating of the likeliness of each hypothesis to which it is relevant. For example, as shown in the left hand side of Fig. 6, the believability of the observations performed by a source (such as Curveball) depends on source's *competence* and *credibility*. Moreover, competence depends on *access* and *understandability*. Credibility depends on *veracity*, *objectivity*, and *observational sensitivity under the conditions of observation*. Thus, the bias that would result from the *persistence of impressions based on discredited evidence* is countered in TIACRITIS with a rigorous, detailed and explicit believability analysis.

But there are additional biases in the evaluation of evidence that Heuer does not mention, particularly with respect to establishing the credentials of evidence: relevance, believability, and inferential force or weight. An analyst may

confuse the competence of a HUMINT source with his/her credibility. Or, the analyst may focus on the veracity of the source and ignore source's objectivity and observational sensitivity. Analysts may fail to recognize possible synergisms in convergent evidence, as happened in the 9/11/2001 disaster. Analysts may even overlook evidence having significant inferential force.

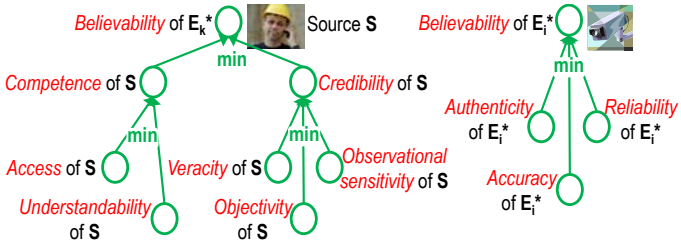


Fig. 6. Believability of testimonial and tangible evidence.

B. *Biases in the Perception of Cause and Effect*

As noted by Heuer, analysts seek explanations for the occurrence of events and phenomena. These explanations involve assessments of causes and effects. But biases arise when analysts assign causal relations to those that are actually accidental or random in nature. One related consequence is that analysts often overestimate their ability to predict future events from past events, because there is no causal association between them. One major reason for these biases is that analysts may not have the requisite level of understanding of the kinds and amount of information necessary to infer a dependable causal relationship.

According to Heuer, when feasible, the "increased use of scientific procedures in political, economic, and strategic research is much to be encouraged", to counter these biases [2, p.128]. Because TIACRITIS makes all the judgments explicit, they can be examined by other analysts to determine whether they contain any mistakes or are incomplete. Because different people have different biases, comparing and debating analyses of the same hypothesis made by different analysts can also help identify individual biases. Finally, as a learning system, TIACRITIS can acquire correct reasoning patterns from expert analysts which can then be used to analyze similar hypotheses.

Now, here is something that can occur in any analysis concerning chains of reasoning. It is always possible that an analyst's judgment will be termed biased or fallacious, on structural grounds if it is observed that this analyst frequently leaves out important links in his/her chains of reasoning. This is actually a common occurrence since, in fact, there is no such thing as a uniquely correct or perfect argument. Someone can always find alternative arguments to the same hypothesis; what this says is that there may be entirely different inferential routes to the same hypothesis. Another possibility is that someone may find arguments based on the same evidence that lead to different hypotheses. This is precisely why there are trials at law; the prosecution and defense will find different arguments, and tell different stories, from the same body of evidence.

C. *Biases in Estimating Probabilities*

There are different views among probabilists on how to assess the force of evidence [1]. The view of probability that Heuer assumes is the conventional view of probability which might be best called the Kolmogorov view of probability since the Russian mathematician was the first one to put this view of probability on an axiomatic basis [13, 14]. This is also the only view of probability considerd by Heuer's sources of inspiration on biases: Daniel Kahneman, Amos Tversky, and their many colleagues in psychology [15, 16]. In his writings, Kolmogorov makes it abundantly clear that his axioms apply only to instances in which we can determine probabilities by counting. But Heuer also notes that intelligence analysis usually deals with one-of-a-kind situations for which there are never any statistics. In such cases, analysts resort to subjective or personal numerical probability expressions. He discusses several reasons why verbal assessments of probability are frequently criticized for their ambiguity and misunderstanding. In his discussion he recalls Sherman Kent's advice that verbal assessments should always be accompanied by numerical probabilities [17].

Since Heuer only considers numerical probabilities conforming to the Kolmogorov axioms, any biases associated with them (e.g., using the availability rule, the anchoring strategy, expressions of uncertainty, assessing the probability of a scenario) are either irrelevant or not directly applicable to a type of analysis that is based on different probability systems, such as the one performed with TIACRITIS, which is based on the Baconian and Fuzzy probability systems. Indeed, analysts using TIACRITIS never assess any numerical probabilities.

Heuer [2, p.122] mentions *coping with evidence of uncertain accuracy* as an origin of bias: "The human mind has difficulty coping with complicated probabilistic relationships, so people tend to employ simple rules of thumb that reduce the burden of processing such information. In processing information of uncertain accuracy or reliability, analysts tend to make a simple yes or no decision. If they reject the evidence, they tend to reject it fully, so it plays no further role in their mental calculations. If they accept the evidence, they tend to accept it wholly, ignoring the probabilistic nature of the accuracy or reliability judgment." He then further notes [2, p.123]: "Analysts must consider many items of evidence with different degrees of accuracy and reliability that are related in complex ways with varying degrees of probability to several potential outcomes. Clearly, one cannot make neat mathematical calculations that take all of these probabilistic relationships into account. In making intuitive judgments, we unconsciously seek shortcuts for sorting through this maze, and these shortcuts involve some degree of ignoring the uncertainty inherent in less-than-perfectly-reliable information. There seems to be little an analyst can do about this, short of breaking the analytical problem down in a way that permits assigning probabilities to individual items of information, and then using a mathematical formula to integrate these separate probability judgments."

First, as discussed in the previous section, concerning the believability of evidence, there is more than just its accuracy to consider. Second, as discussed above, Heuer only considers the *conventional view of probability* which, indeed, involves complex probability computations. With TIACRITIS, the analyst does precisely what Heuer imagined that could be done for countering this bias. It breaks a hypothesis into simpler hypotheses (see Fig.1), and assesses the simpler hypotheses

based on evidence (see Fig.5). Also, TIACRITIS allows the analyst to express probabilities in words rather than numbers, and to employ simple min/max strategies for assessing the probability of interim and final hypotheses that do not involve any full-scale and precise Bayesian or other methods that would require very large numbers of probability assessments.

There are many places to begin a defense of verbal or fuzzy probability statements. The most obvious one is law. All of the forensic standards of proof are given verbally: "beyond reasonable doubt"; "clear and convincing evidence", "balance of probabilities"; "sufficient evidence", and "probable cause'. Over the centuries attempts have been made to supply numerical probability values and ranges for each of these standards, but none of them have been successful. The reason, of course, is that every case is unique and rests upon many subjective and imprecise judgments. Wigmore [18] understood completely that the catenated inferences in his Wigmorean networks were probabilistic in nature. Each of the arrows in the chain of reasoning describe the force of one hypothesis on the next one, e.g., E → F. Wigmore graded the force of such linkages verbally using such terms as "strong force", "weak force", "provisional force", etc. Toulmin [19] also used fuzzy qualifiers in the probability statements of his system which grounds Rationale [20]. There are many other examples of situations in which it is difficult or impossible for people to find numerical equivalents for verbal probabilities they assess. Intelligence analysis so often supplies very good examples in spite of what Sherman Kent said some years ago.

We conclude this discussion by recalling what the well-known probabilist Professor Glenn Shafer said years ago [21]: *Probability is more about structuring arguments than it is about numbers. All probabilities rest upon arguments. If the arguments are faulty, the probabilities however determined, will make no sense.* In TIACRITIS, the structure of the bottom-up argument is given by the logical top-down decomposition, and the conclusions are hedged by employing rigorous Baconian operations with fuzzy qualifiers, leading to a defensible and persuasive argument.

D. Hindsight Biases in Evaluating Intelligence Reporting

As Heuer notes, analysts often overestimate the accuracy of their past judgments; customers often underestimate how much they have learned from an intelligence report; and persons who conduct post-mortem analysis of an intelligence failure will judge that events were more readily foreseeable than was in fact the case. "The analyst, consumer, and overseer evaluating analytical performance all have one thing in common. They are exercising hindsight. They take their current state of knowledge and compare it with what they or others did or could or should have known before the current knowledge was received. This is in sharp contrast with intelligence estimation, which is an exercise in foresight, and it is the difference between these two modes of thought—hindsight and foresight—that seems to be a source of bias. … After a view has been restructured to assimilate the new information, there is virtually no way to accurately reconstruct the pre-existing mental set." [2, p.162]

Apparently Heuer did not envision the use of a system like TIACRITIS that keeps track of the performed analysis, what evidence we had, what assumptions we made and what were their justifications, and what was the actual logic of our analytic conclusion. We can now add additional evidence and use our hindsight knowledge to restructure the argumentation and re-evaluate our hypotheses, and we can compare the hindsight analysis with the foresight one. But we will not confuse them. As indicated by Heuer [2, pp.166-167]: "A fundamental question posed in any postmortem investigation of intelligence failure is this: Given the information that was available at the time, should analysts have been able to foresee what was going to happen? Unbiased evaluation of intelligence performance depends upon the ability to provide an unbiased answer to this question.**"** We suggest that this may be accomplished with a system like TIACRTIS.

IV. Some Frequently Overlooked Origins of Bias

So much of the discussion of bias in intelligence analysis is directed at intelligence analysts themselves. But we have identified three other origins of bias that are rarely discussed, even though they may be at least as important on occasion as any analysts' alleged biases. The three other origins of bias we will consider are: (1) persons who provide testimonial evidence about events of interest (i.e. HUMINT sources); (2) other intelligence professionals having varying capabilities who serve as links in what we term "chains of custody" linking the evidence itself, as well as it's sources, with the users of evidence (i.e. the analysts); and (3) the "consumers" of intelligence analyses (government and military officials who make policy and decisions regarding national security).

A. HUMINT Sources

Our concern here is with persons who supply us with testimonial evidence consisting of reports of events about matters of interest to us. Heuer [2, p.122] does mention the "bias on the part of the ultimate source," but he does not analyze it. In our work on evidence in a variety of contexts, we have always been concerned about establishing the believability of its sources, particularly when they are human witnesses, sources, or informants [1]. In doing so, we have made use of the 600 year-old legacy of experience and scholarship in the Anglo-American adversarial trial system concerning witness believability assessments. We have identified the three major attributes of the credibility of ordinary witnesses: *veracity*, *objectivity*, and *observational sensitivity* (see Fig. 6). We will show how there are distinct and important possible biases associated with each such believability attribute. These biases are recognized in the MACE system (Method for Assessing the Credibility of Evidence), developed for the IC [22]. This system incorporates both Baconian and Bayesian methods for combining evidence about our source.

As discussed above, assessing the credibility of a human source **S** involves assessing **S**'s veracity, objectivity, and observational sensitivity. We have to consider that source **S** can be biased concerning any of these attributes. On *veracity*, **S** might prefer to tell us that event **E** occurred, whether **S** believed **E** occurred or not. As an example, an analyst evaluating **S**'s evidence **E*** might have evidence about **S** suggesting that **S** would tell us that **E** occurred because **S** wishes to be the bearer of what **S** believes we will regard as good news that event **E** occurred. On *objectivity*, **S** might choose to believe that **E**

occurred because it would somehow be in **S**'s best interests if **E** did occur. On *observational sensitivity*, there are various ways that **S**'s senses could be biased in favor of recording event **E**; clever forms of deception supply examples.

These three species of bias possible for HUMINT sources must be considered by analysts attempting to assess the credibility of source **S** and how much weight or force **S**'s evidence **E*** should have in the analyst's inference about whether or not event **E** did happen. The existence of any of these three biases would have an effect on an analyst's assessment of the weight or force of **S**'s report **E***. As we know, all assessments of the credibility of evidence rest upon available evidence about its sources. In the case of HUMINT we need ancillary evidence about the veracity, objectivity, and observational sensitivity of its sources. In the process, we have to see whether any such evidence reveals any of the three biases just considered. TIACRITIS supports the analyst in this determination by guiding her to answer specific questions based on ancillary evidence. For instance, the veracity questions considered are shown in Table 1.

Table 1. Questions concerning the veracity of human sources.

1. *Goals of this source?* Does what this source tells us support any of his or her goals?
2. *Present influences on this source?* Could this source have been influenced in any way to provide us with this report?
3. *Exploitation potential?* Is this source subject to any significant exploitation by other persons or organizations to provide us this information?
4. *Any contradictory or divergent evidence?* Is there any evidence that contradicts or conflicts with what the source has reported to us?
5. *Any corroborative or confirming evidence?* Is there any other evidence that corroborates or confirms this source's report?
6. *Veracity concerning collateral details?* Are there any contradictions or conflicts in the collateral details provided by this source that reflect the possibility of this source's dishonesty?
7. *Source's character?* What evidence do we have about this source's character and honesty that bears upon this source's veracity?
8. *Reporting record?* What does the record show about the truthfulness of this source's previous reports to us?
9. *Source expectations about us?* Is there any evidence that this source may be reporting events he/she believes we will wish to hear or see?
10. *Interview behavior?* If this source reported these events to us, what was this source's demeanor and bearing while giving us this report?

B. *Persons in Chains of Custody of Evidence*

Unfortunately, there are other persons, apart from HUMINT sources, whose possible biases need to be carefully considered. We know that analysts make use of an enormous variety of evidence that is not testimonial or HUMINT, but is *tangible* in nature. Examples include objects, images, sensor records of various sorts, documents, maps, diagrams, charts, and tabled information of various kinds.

But the intelligence analysts only rarely have immediate and first access to HUMINT assets or informants. They may only rarely be the first ones to encounter an item of tangible evidence. What happens is that there are several persons who have access to evidence between the times the evidence is first acquired and when the analysts first receive it. These persons may do a variety of different things to the initial evidence during the time they have access to it. In law, these persons constitute what is termed a "*chain of custody*" for evidence.

Heuer [2, p.122] mentions the "distortion in the reporting chain from subsource through source, case officer, reports officer, to analyst" but he does not analyze it. In criminal cases in law, there are persons identified as "evidence custodians", who keep careful track of who discovered an item of evidence, who then had access to it and for how long, and what if anything they did to the evidence when they had access to it.

These chains of custody add three major additional sources of uncertainty for intelligence analysts to consider, that are associated with the persons in chains of custody whose competence and credibility need to be considered. The first and most important question involves *authenticity*: *Is the evidence received by an analyst exactly what the initial evidence said and is it complete?* The other questions involve assessing the *reliability* and *accuracy* of the processes used to produce the evidence if it is tangible in nature (see the right side of Fig. 6), or also used to take various actions on the evidence in a chain of custody, whether the evidence is tangible or testimonial. As an illustration, consider an item of testimonial HUMINT coming from a foreign national whose code name is "Wallflower", who does not speak English [23]. Wallflower gives his report to *case officer* Bob. This report is *recorded* by Bob and then *translated* by Husam. Then, Wallflower's translated report is *transmitted* to a *report's officer* Marsha who *edits* it and *transmits* it to the analyst Clyde who evaluates it and assesses its weight or force.

Now, here is where forms of bias can enter that can be associated with the persons involved in these chains of custody. The case officer Bob might have intentionally overlooked details in his recording of Wallflower's report. The translator Husam may have intentionally altered or deleted parts of this report. The report's officer Marsha might have altered or deleted parts of the translated report of Wallflower's testimony in her editing of it. The result of these actions is that the analyst Clyde receiving this evidence almost certainly did not receive an authentic and complete account of it, nor did he receive a good account of its reliability and accuracy. What he received was the transmitted, edited, translated, recorded testimony of Wallflower. Fig. 7 shows how TIACRITIS may determine the believability of the evidence received by the analyst. Although the information to make such an analysis may not be available, the analyst should adjust the confidence in his conclusion, in recognition of these biases.

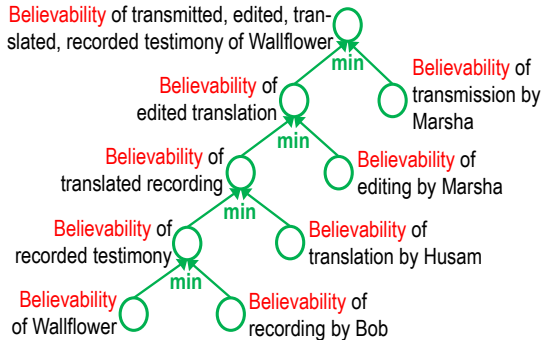


Fig. 7. Chain of custody of Wallflower's testimony.

C. *Consumers of Intelligence Analyses*

The policy-making consumers or customers of intelligence

analysts are also subject to a variety of inferential and decisional biases that may influence the reported analytic conclusions. As is well known, the relationships between intelligence analysts and governmental policy makers are much discussed and involve considerable controversy [24, 25]. On the one hand we hear intelligence professionals say that they do not make policies but only try to help policy makers be as informed as they can be when they do form policies and make decisions in the nation's best interests. But we also learn facts about the intelligence process that complicate matters. An intelligence analysis is usually a hierarchical process involving many intelligence officers, at various grade levels, who become involved in producing an intelligence "product". At the most basic level of this hierarchy are the so-called "desk analysts" who are known and respected experts in the specific subject matter of the analysis at hand. An analysis produced by one or more desk analysts is then passed "upward" through many administrative levels, at each of which persons at these higher levels can comment on the desk analysts' report. It is often recognized that the higher an editor is in this hierarchy, the more political his/her views and actions become that may affect the content and conclusions of the analysis at hand. As this "upward" process continues, the analysis that results may be quite different from the one produced by the desk analysts, reflecting the biases of those who have successively edited it. In some cases, these editing biases are the direct result of the consumer's biases who may wish to receive a certain analytic conclusion. Using a system like TIACRITIS that shows very clearly how the analytic conclusion is rooted in evidence would significantly help in reducing the above biases.

V. CONCLUSIONS

A wide variety of biases affect the correctness of intelligence analyses. In this paper we have shown how the use of TIACRITIS, a knowledge-based cognitive assistant, helps analysts recognize and counter many of them. TIACRITIS integrates several semantic technologies (knowledge representation through ontologies and rules, evidence-based reasoning, machine learning and knowledge acquisition). It can run in a browser as a web-based system, or it can be installed locally, and has been used in many civilian, military, and intelligence organizations.

There are two complementary ways by which TIACRITIS helps mitigate biases. First, as a cognitive assistant, it helps automate many parts of the analysis process, making this task much easier for the analyst. Thus it alleviates one of the main causes of biases, which is the employment of simplified information processing strategies on the part of the analyst. Second, TIACRITIS performs a rigorous evidence-based hypothesis analysis that makes explicit all the reasoning steps, evidence, probabilistic assessments, and assumptions, so that they can be critically analyzed and debated. Indeed, the best protection against biases comes from the collaborative effort of teams of analysts, who become skilled in solving their analytic tasks through the development of sound evidence-based arguments, and who are willing to share their insights with colleagues, who are also willing to listen. TIACRITIS makes all this possible.

Finally, this paper adds a strong argument in favor of using structured analytic methods, in the debate on how to significantly improve intelligence analysis [26].

REFERENCES

[1] Schum D.A. (2001). *The Evidential Foundations of Probabilistic Reasoning,* Northwestern University Press.

[2] Heuer R.J. (1999). *Psychology of Intelligence Analysis*, Center for the Study of Intelligence, Central Intelligence Agency, Washington, DC.

[3] Tecuci, G., Marcu, D., Boicu, M., Schum, D.A., Russell K. (2011). Computational Theory and Cognitive Assistant for Intelligence Analysis, in *Proc. 6th Int. Conf. on Semantic Technologies for Intelligence, Defense, and Security*, pp. 68-75, Fairfax, VA, 16-18 November.

[4] Cohen L.J. (1977). *The Probable and the Provable,* Clarendon Press, Oxford.

[5] Zadeh L. (1983). The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems, *Fuzzy Sets and Systems*, vol.11, pp.199-227.

[6] Tecuci, G. (1998). *Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies,* San Diego: Academic Press.

[7] Tecuci, G., Boicu, M., Boicu, C., Marcu, D., Stanescu, B., Barbulescu, M. (2005). The Disciple-RKF Learning and Reasoning Agent, *Computational Intelligence, Vol.21, No.4*, pp. 462-479.

[8] Tecuci, G., Boicu, M., Marcu, D., Schum, D. (2013). How Learning Enables Intelligence Analysts to Rapidly Develop Practical Cognitive Assistants, in *Proc. 12th International Conference on Machine Learning and Applications* (ICMLA'13), Miami, Florida, December 4-7.

[9] W3C (2004). http://www.w3.org/TR/rdf-schema/, accessed 10/11/13.

[10] Boicu M., Tecuci G., Schum D. (2008). Intelligence Analysis Ontology for Cognitive Assistants, in *Proc. of Conf. "Ontology for the Intelligence Community"*, Fairfax, VA, 3-4 December.

[11] Heuer R.J. (2008). Computer-Aided Analysis of Competing Hypotheses, in George R.Z., Bruce J.B., eds., *Analyzing Intelligence: Origins, Obstacles, and Innovations*, Georgetown Univ. Press, Washington, DC.

[12] Drogin B. (2007). *CURVEBALL: Spies, Lies, and the Con Man Who Caused a War*. Random House, New York, NY.

[13] Kolmogorov A.N. (1933). *Foundations of a Theory of Probability* (1933), 2nd English edition, Chelsea Publishing, New York, NY., 1956

[14] Kolmogorov A.N. (1969). The Theory of Probability. In: Aleksandrov, A. D., Kolmogorov, A. N., Lavrentiev, M. A. (eds) *Mathematics: Its Content, Methods, and Meaning*. MIT Press, Cambridge, MA.

[15] Kahneman, D., Tversky, A. (1974). Judgment under Uncertainty: Heuristics and Biases. *Science*, Vol, 185, 1124-1131.

[16] Kahneman, D., Slovic, P., Tversky, A. (1982). *Judgment under Uncertainty: Heuristics and Biases.* Cambridge University Press.

[17] Kent S. (1994). Words of Estimated Probability, in Steury D.P., ed., *Sherman Kent and the Board of National Estimates: Collected Essays*, Center for the Study of Intelligence, CIA, Washington, DC.

[18] Wigmore J.H. (1937). *The Science of Judicial Proof.* Boston, MA: Little, Brown & Co.

[19] Toulmin S.E. (1963). *The Uses of Argument*. Cambridge Univ. Press.

[20] van Gelder T.J. (2007). The Rationale for Rationale, *Law, Probability and Risk*, 6, pp. 23-42.

[21] Shafer G. (1988). Combining AI and OR. *University of Kansas School of Business Working Paper No. 195*.

[22] Schum D.A., and Morris J. (2007). Assessing the Competence and Credibility of Human Sources of Evidence: Contributions from Law and Probability, *Law, Probability and Risk,* Vol. 6, pp. 247-274.

[23] Schum D.A., Tecuci G, and Boicu M. (2009). Analyzing Evidence and its Chain of Custody: A Mixed-Initiative Computational Approach, *Int. Journal of Intelligence and Counterintelligence,* Vol. 22, pp. 298-319.

[24] George, R., Bruce, J. (eds) (2008). *Analyzing Intelligence: Origins, Obstacles, and Innovations.* Georgetown Univ. Press, Washingon. DC.

[25] Johnston, R. (2005). *Analytic Culture in the U.S. Intelligence Community*. Central Intelligence Agency, Washington, DC.

[26] Marrin, S. (2011). *Improving Intelligence Analysis: Bridging the gap between scholarship and practice.* Routlege, Lundon and New York.

IAO-Intel

An Ontology of Information Artifacts in the Intelligence Domain

Barry Smith
University at Buffalo
NY, USA

Tatiana Malyuta
CUNY, NY, USA
Data Tactics, McLean, VA

Ron Rudnicki
CUBRC, Buffalo
NY, USA

William Mandrick
Data Tactics
McLean, VA, USA

David Salmen
Data Tactics
McLean, VA, USA

Peter Morosoff
E-Maps, Inc.
Washington, DC, USA

Danielle K. Duff
I2WD
Aberdeen, MD, USA

James Schoening
I2WD
Aberdeen, MD, USA

Kesny Parent
I2WD
Aberdeen, MD, USA

***Abstract*—We describe on-going work on IAO-Intel, an information artifact ontology developed as part of a suite of ontologies designed to support the needs of the US Army intelligence community within the framework of the Distributed Common Ground System (DCGS-A). IAO-Intel provides a controlled, structured vocabulary for the consistent formulation of metadata about documents, images, emails and other carriers of information. It will provide a resource for uniform explication of the terms used in multiple existing military dictionaries, thesauri and metadata registries, thereby enhancing the degree to which the content formulated with their aid will be available to computational reasoning.**

Keywords—ontology; information artifacts; military doctrine; intelligence analysis*; *interoperability*; *data services environment

I. BACKGROUND

Standardization of terminology has been important from the very beginning of organized warfare. Imagine the Chinese trying to pass reports down the Great Wall using fire beacons without standardization of the signals used. In the Revolutionary War, General Washington directed Friedrich Wilhelm von Steuben to write the drill manual for the Continental Army [1] so that all units would use and respond uniformly to the same commands.

In our own era, DoD has directed development and use of the *DoD Dictionary of Military and Associated Terms* (Joint Publication 1-02) as the paramount terminological standard for military operations [2]. JP 1-02 helps to enable joint warfare by (a) advancing consistency in communications and (b) facilitating consistent interpretation of commands. Military dictionaries and related terminology artifacts continue to be developed, addressing these and a series of additional aims, including: (c) compiling lessons learned (outcomes assessment); (d) providing controlled vocabularies for official reporting; and (e) enhancing discoverability and analysis of data.

Such artifacts have until recently been conceived by analogy with traditional free-text dictionaries published in forms designed to maximize utility to human beings. Most existing doctrinal and related lexica and thesauri not only provide little aid to computation, they also suffer from the fact that multiple such resources have been (and continue to be) developed independently, in divergent and often non-principled ways. The result is that identical data may be classified and described entirely differently by different agencies, and the consequences of the resultant failures of integration (for example in the case of registries of persons of interest) are all too familiar. Increasingly, however, it is recognized that there is the need for a unified approach to description and classification of information resources (see for example [3], [4]), and the DoD has recognized at an official level that, to advance discoverability and analysis in the age of Big (military) Data, new approaches are needed that can enable computational retrieval, integration and processing of data. Thus Directive 8320.02 [5], the latest version of which is dated August 5, 2013, requires all authoritative DoD data sources to be registered in the DoD Data Services Environment (DSE) [6]. It further requires that all salient metadata be discoverable, searchable, retrievable, and understandable:

> Data, information, and IT services will be considered understandable when authorized users are able to consume them and when users can readily determine how those assets may be used for specific needs. Data standards and specifications that require associated semantic and structural metadata, including vocabularies, taxonomies, and ontologies, will be published in the DSE, or in a registry that is federated with the DSE.

We shall return to the DSE below. First, we present our own strategy for realizing these important goals.

II. THE INFORMATION ARTIFACT ONTOLOGY

The Information Artifact Ontology (IAO) was originally conceived in 2008 as part of an effort to master the Big Data accumulating in the wake of the Human Genome Project in the context of biological research [7]. Its goal was to aid the consistent description of biological data emanating from multiple heterogeneous sources. The goal of IAO-Intel is analogous: it is to provide common resources for the consistent description of information artifacts of relevance to the intelligence community in a way that will allow discovery, integration and analysis of intelligence data from both official and non-official sources.

When biomedical informaticians work with databases, publications and records generated by experimental research or medical care they focus primarily on what these artifacts *describe* (for example on the genes or proteins which form the subject matters of a given journal publication, or on the symptoms or diseases reported in a given clinical note). Similarly, when intelligence analysts work with source data artifacts, then they, too, focus primarily on what the data in these artifacts describe, for example on the military units

whose movements are recorded in a given shipping report, or on the vulnerabilities of a given forward operations base as described in some force protection assessment.

But while the primary focus concerns in both cases the topic or subject of the artifacts in question, both also require a secondary focus, targeted to the artifacts themselves, through which information about these topics is conveyed. Such artifacts have attributes – including format, purpose, evidence, provenance, operational relevance, security markings – data concerning which (often called 'metadata') is vital to the effective exploitation of the reports, images, or signals documents with which the analyst has to deal.

The dichotomy between *focus on entities in the world* and *focus on the information artifacts in which these entities are represented* is fundamental to the work reported here. IAO relates precisely to the objects of this secondary focus. An information artifact (IA), as we conceive it, is an entity that has been created through some deliberate act or acts by one or more human beings, and which endures through time, potentially in multiple (for example digital or printed) copies. IAO thus deals with information in the forms it takes when it has been deliberately fixed in some medium in such a way as to become accessible to multiple subjects. Examples are: a diagram on a sheet of paper, a video file, a map on a computer monitor, an article in a newspaper, a message on a network, the output of some querying process in a computer memory.

III. GOAL OF IAO-INTEL

The goal of IAO-Intel is to support the effective handling of data concerning those attributes of IAs that are relevant to the purposes of intelligence analysis. To describe such attributes coherently we need to distinguish:

– the *particular* information artifact of interest, tied to some particular physical information bearer: the photographic image on this piece of paper retrieved from this enemy combatant; the email created by this particular author on this specific laptop; the target list compiled for this particular artillery unit on this particular date;

– the *copyable* information content that is carried by the artifact in question. The photographic image may be printed out in multiple paper copies; the email or target list may be transmitted to multiple further recipients. The information content that is copied or transmitted thereby remains in each case one and the same.

IAO-Intel provides ontology terms relating both to official documents and to non-official (source) artifacts. It provides also a set of relations to be used when we wish to represent the fact that, say, IA #12345 *is-about* some given person, or *uses-symbols-from* some specified symbology, or *links-to* some second IA #56789, and so forth,

IAO-Intel is designed from the start to provide the needed supplement in a way that will create semantic interoperability of data retrieved from different types of sources through an incremental process of semantic enhancement as described in [8], [9] and [10]. It is designed to allow automatic retrieval of all documents in a given collection of heterogeneous sources which involve a particular creator, or a particular type of intelligence report, or a particular type of weblink, or have been declassified under the authority of a particular agency, or are operative within a given time window.

Importantly, IAO-Intel is not designed to *replace* existing doctrinal or other standards created to guide human beings or computer applications in the creation and description of documents in accordance with defined formats or document architectures. Rather, its purpose is to allow the results of using such standards to generate the needed metadata in a uniform, non-redundant and algorithmically processable fashion. Moreover, the broad scope of IAO-Intel means that the metadata generated in relation to *official* documents will be of a piece with the metadata incrementally accumulating in relation to all information artifacts of relevance to the IC – the metadata will consist, in every case, of annotations to IAs formulated in ontology terms drawn not only from IAO-Intel but from the entire suite of DSGS-A ontology modules.

Thus while using existing standards for human or computer-aided creation or description of IAs does indeed allow us to retrieve data pertaining to IAs prepared in accordance with these standards, for IAs of other sorts the existing approach will fail. Only an ontology-based approach along the lines here proposed can, we believe, demonstrate the sort of flexibility and consistent expandability which are needed in today's dynamic and data-rich environments.

IV. EXPLICATION AND ANNOTATION

Currently a draft version of IAO-Intel is being applied within the framework of the US Army's Distributed Common Ground System (DCGS-A) Standard Cloud (DSC) initiative as part of a strategy for the horizontal integration of warfighter intelligence data [9]. Two sorts of application are currently being used to enable the ontology to support computer-aided retrieval and analytics. First, is *explication* of general terms used in source intelligence artifacts and in data models, terminologies and doctrinal publications which provide typologies of intelligence-related IAs. Second, is the *annotation* of the instance-level information captured by such IAs.

Explication is performed by providing definitions of such *general* terms using the resources of IAO-Intel and of the domain ontologies (such as Agent or Event ontologies) being developed within the DSGS-A framework. Annotation is performed by associating ontology terms with data about *particular* persons, events, or places in given information artifacts.

TABLE 1. SAMPLE TYPES AND SUBTYPES OF INFORMATION ARTIFACTS

IAO	IAO-Intel (examples)
Report	Intelligence Report (FM 6-99.2, 126)
Summary	Electronic Warfare Mission Summary (FM 6-99.2, 87)
Diagram	Network Analysis Diagram (from JP 2-01.3, II-51)
Overlay	Combined Information Overlay (JP 2-01.3, II 33)
Assess-ment	Assessment of Impact of Damage (FM 6-99.2, 53)
Estimate	Adversary Course of Action Estimate
List	List of High-Value Targets (JP 2-01.3, II 61)

Order	Airspace Control Order (FM 6-99.2, 17)
Matrix	Target Value Matrix (JP 2-01.3, II-63)
Template	Ground and Air Adversary Template (JP 2-01.3, II-57)

The goal of explication is to ensure that the data captured in annotations is semantically enhanced in a way that enables computational integration and reasoning along the lines described in [11], [12]. The goal of annotation is to aid retrieval of information about specific persons, groups, events, documents, images, and so forth, where this information is conveyed through source documents using disjointed and disparate systems for designation.

V. Strategy for Building IAO-Intel

Our strategy for building IAO-Intel is to extend the draft IAO to include terms and definitions tailored for the intelligence domain and specifically for the needs of our DSGS-A ontology initiative. The strategy has the following parts.

First, IAO-Intel is created by downward population from the draft IAO reference ontology. That is, the highest level terms of IAO-Intel are defined as specializations of terms from IAO along the lines illustrated in Table 1. The coverage domain of IAO-Intel will be determined incrementally on the basis of requests from analysts and other SME communities and through incorporation of terms from doctrinal publications and relevant high-level data models and document classifications.

Second, we use these sources to identify the dimensions of attributes along which IAs will be annotated. The selected dimensions are constructed in such a way as to be *orthogonal* in the sense in which, for example, color is orthogonal to shape – thus ontology branches built to represent different dimensions of attributes will contain no terms in common. This will enable these branches to be structured following the principle of single inheritance (thus as true hierarchies) [13].

Third, we create *low-level ontology modules* (LLOs) corresponding to each of these orthogonal dimensions. LLOs are small single-dimension attribute lists or shallow hierarchies designed to advance ease of maintenance and surveyability of the ontology and to provide a growing set of simple component terms which can be used:

1. to *construct more complex terms*, both terms for inclusion in IAO-Intel, and terms to be used to generate inferred classifications in application ontologies created for specific local purposes, along the lines described in [10];
2. to *define* the terms of the IAO-Intel ontology and of its sister ontologies within the DSGS-A framework;
3. to *explicate* the meanings of terms standardly used by different agencies, or by different groups of SMEs, or by different existing and future systems to describe such artifacts in a logically consistent way that is designed to allow integration of data and enhanced analytics;
4. to *annotate* instance data pertaining to particular information artifacts used by the intelligence community – for instance analysts' reports; harvested emails; signals data; and so forth.

The goal is that IAO-Intel should support integration of data annotated using different standard terminology resources. To bring this about, the constituent terms of such resources will be explicated using terms from IAO-Intel so that the artificial composite terms used in certain official terminologies and exchange model resources (along the lines of 'VehicleInspectionJurisdictionAuthorityText') will be broken down logically into constituent elements. This will provide a means to avoid the combinatoric explosion that is threatened by traditional approaches. Some composite expressions – for example 'Essential Element of Friendly Information (EEFI)' – will indeed be included in pre-composed form in the IAO-Intel ontology, but only where they are either defined in doctrine or already established as part of relevant SME vocabularies.

The modeling task for which compounds such as 'VehicleInspectionJurisdictionAuthorityText' were designed is addressed in our framework by allowing single data entries to be annotated by multiple ontology terms (sometimes linked by appropriate relations). A record in one of the tables containing data about an IED can be annotated, for example, both with 'IED Event' (based on its aboutness) and with 'EEFI' (based on its importance). A particular plan for the Intelligence Preparation of the Battlefield can be annotated as being at the same time a Plan (based on its purpose), a Government Document (based on its source), a Report on Air Defenses (based on its aboutness). It can be annotated also through relations, for example through *located-at* linking the source of the plan to some city or building and linking the planned air defenses to some region of interest.

Currently, military terminology resources generally fail to follow established best practice principles for the formulation of definitions. For example, they often confuse terms referring to components of information artifacts with terms referring to the entities in reality which those information artifacts are about. The "WTI Improvised Explosive Device" Glossary, for example, defines Method of Emplacement as:

> The description of where the [improvised explosive] device was delivered, used or employed.

Similarly the DCGS-A Logical Data Model defines Cover-Concealment as:

> information about geographical features that provide protection from attack or observation.

Use of IAO-Intel in tandem with corresponding domain ontologies allows us to explicate CoverConcealment (properly so-called) as:

> a geographic feature which *has-role* CoverRole,

and to explicate CoverConcealmentInformation as:

> IA which *is-about* CoverConcealment,

where CoverRole is defined as:

> the Role acquired by a given geographic feature when it is used to provide protection from attack or observation.

VI. Maintaining and Evaluating IAO-Intel

To maintain the IAO-Intel term collection over time we will create feedback links to enable users of the ontology to request new terms and to report errors. We are also working

on an objective validation process which will enable us to determine how requested terms should be treated, distinguishing options such as: 1. incorporation into IAO-Intel or into some associated reference ontology, 2. incorporation into an application ontology maintained for some local purpose, 3. being marked as a synonym of some existing ontology term.

We are identifying, and where necessary constructing *de novo*, the domain ontologies that will need to be used in the definition of complex terms, and defining the relations that will link IAO-Intel terms with terms in these domain ontologies. These ontologies, too, will be extended over time on the basis of input from users.

We are also testing a series of objective criteria to be used in evaluation of IAO-Intel and other DCGS-A ontologies, starting with simple numerical measures of (a) term requests received and dealt with, and (b) uses of terms in *definitions*, *explications* and *annotations*. IAO-Intel will allow us to keep track of the number of information artifacts that make reference to individuals falling under a given class, and these metrics too can be used to assess the relative importance of this class within the ontology framework taken as a whole. While not definitive, such measures will help guide our judgments concerning the content and structure both of IAO-Intel and of its associated domain ontologies.

VII. ORGANIZATION OF IAO-INTEL

Given the importance of the dichotomy between primary (topic) and secondary (artifact) focus, a central role in IAO-Intel is played by what we call

– Information Content Entities (ICEs) are *about* something in reality (they have this something *as a subject*; they *represent*, or *mention* or *describe* this something; they *inform us about* this something). Aboutness may be identifiable from different perspectives. Thus one analyst may interpret a given ICE as being about the geography of a given encampment; another may view it as providing information about the morale of those encamped there.

All major classes of information artifacts involve ICEs – simply because all major classes of information artifacts are about something. A plan of action, for example, is *about* a certain group of persons and goals and the types and ordering of actions that will be used to realize these goals. Even a document that has been written in code will be assumed by an analyst to be about something (for what, otherwise, would be the reason for its creation?). Typically, an information artifact such as a copy of a newspaper will be associated with multiple ICEs at successive levels of granularity, including separate articles within the newspaper, separate sentences within these articles, and so on.

In addition to ICEs, we distinguish also:

– *Information Bearing Entity* (IBE). An IBE is a material entity that has been created to serve as a bearer of information. IBEs are either (1) self-sufficient material wholes, or (2) proper material parts of such wholes. Examples under (1) are: a hard drive, a paper printout (e.g., a report); and under (2): a specific sector on a hard drive, a single page of a paper printout.

– *Information Quality Entity* (IQE). An IQE is the pattern on an IBE in virtue of which it is a bearer of some information.

– *Information Structure Entity* (ISE). An ISE is a structural part of an ICE; speaking metaphorically, it is an ICE with the content removed: for example an empty cell in a spreadsheet; a blank Microsoft Word file. ISEs thus capture part of what is involved when we talk about the 'format' of an IA.

The term 'information artifact' can now be used to refer either 1. to some combination of ICEs and ISEs (roughly: the IA as body of copyable information content); or 2. to some concretization of ICEs and ISEs in some IBE in which some IQE inheres (the information artifact is: this content here and now, on this specific computer screen or this printed page). Different information artifact types will differ in different ways along these dimensions, as illustrated in Table 2.

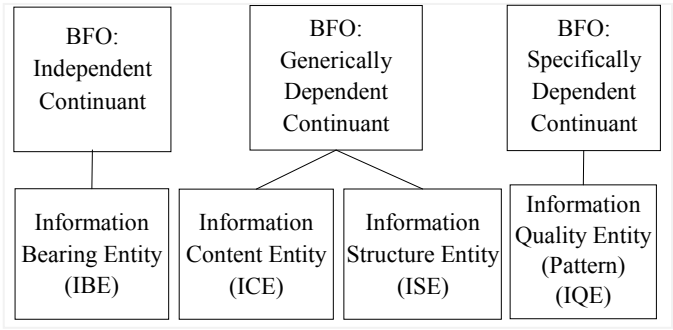


Figure 1. Continuants in the IAO framework

VIII. IAO AND THE BASIC FORMAL ONTOLOGY

Figure 1 shows how IAO and IAO-Intel are being built to conform to Basic Formal Ontology (BFO), the upper-level architecture used in the DSGS-A ontologies [14]. IBEs are, in BFO terms, *independent continuants* (they are entities made of physical matter). An IBE is a physical entity that is created or modified to serve as bearer of certain patterned arrangements – for example of ink or other chemicals, of electromagnetic excitations. An IQE is a quality of an IBE which exists in virtue of such patterned arrangements and which is interpretable as an ICE or ISE. Such an IQE is created when some physical artifact is deliberately created or modified to support it (patterned to serve as its bearer). IQEs are BFO:*specifically dependent continuants* (SDCs) – entities which require some specific physical bearer but which are not themselves physical. Each IBE and IQE is restricted at any given time to some specific location in space. (If you display the same digital image twice on your desktop, then there are two IQEs on your desktop, which are – at some level of granularity – indistinguishable copies of each other.

ICEs and ISEs, in contrast, are what BFO calls *generically dependent continuants* or GDCs. This means that they are entities – such as a pdf file or an email – which can be copied from one physical bearer to another and thus may exist simultaneously in multiple different IQEs, which are called 'concretizations' of the corresponding GDC. Each GDC is concretized by at least one specific IQE inhering for example in the tiny piles of ink on the piece of paper in your pocket or in differentially excited pixels on your screen. When the GDC

is copied, then a new IQE is created on a new physical information bearer, as when a new pattern of characters is created on the screen of the recipient of an email. This second pattern is a copy of the pattern created on the screen of the sender. The GDC itself exists simultaneously both at its original site and at the site to which it has been transmitted. GDCs can thus be multiply located.

BFO *relations* between ICEs, ISEs, IQEs and IBEs can be set forth as follows:

ICE *generically-depends-on* IBE
ISE *generically-depends-on* IBE
IQE *specifically-depends-on* IBE
ICE *concretized-by* IQE
ISE *concretized-by* IQE

IAO contains in addition relations which allow us to formulate metadata concerning attributes of IAs such as author, creation date, classification status, and so forth, and to annotate also components of IAs such as the To- and FromAddress components of email headers. The ToAddress of email message *m*, for example, is defined as:

> a collection of at least one email addresses of the intended recipients of *m*, each with at most one optionally associated name.

The set of relations can be extended to include also relations involving documents, document parts and document collections, such as *retrieved-from*, *curated by*, and so forth.

When we consider examples such as those provided in Table 2, then it becomes clear that, when IAO-Intel is applied to the explication of terms involved in describing instance-data relating to real-world IAs, then multiple artifacts may need to be distinguished. Consider, for example, a pdf file stored on some specific laptop. When we address what is meant by the (copyable) content of this file, then we recognize that this content may be copied in multiple ways, for example: to a pdf file using the same version of the Acrobat software and on the same operating system, to a pdf file using a different version of the Acrobat software, using characters from the same or a different character set, by being printed out on a piece of paper, and so on. The annotation of instance data with information of this sort may be important for example in investigating the provenance of given information artifacts which lie at the end of long chains of copying and processing involving multiple authors and computer systems. One potential application of IAO-Intel is to the systematic annotation of data pertaining to such chains.

Matters are complicated further when we go deeper into the question of how IAs are stored inside the computer. Given a generically dependent continuant which is the pdf file stored in the hard drive on some given laptop, there is a specifically dependent IQE which is (roughly) the pattern of 1s and 0s in the magnetic coating of the hard drive. When the entirety of this pdf file is displayed on your screen, then there is a further specifically dependent IQE which is the corresponding pattern of pixels on your screen. Both of these IQEs are concretizations of a corresponding GDC.

Note that we do not assume that all portions of IAO-Intel will be of equal utility in applications for the IC. We do, however, believe that to achieve clarity of explication in the treatment of source data artifacts will require clear definitions of the upper-level terms in the IAO, and a clear understanding of the relations between them.

TABLE 2: DIMENSIONS OF INFORMATION ARTIFACTS (IAS)

Information Artifact	**IBE**	**ISE**	**ICE**
MS Word file (.doc, .docx)	Hard drive (magnetized sector)	MS Word format	Varies
XML file	Hard drive (magnetized sector)	XML V 2.0 format	Varies
MS Excel 2010 file (.xls, .xlsx)	Hard drive (magnetized sector)	MS Excel 2010 format	Varies
KML file	Hard drive (magnetized sector)	KML	Map overlay
JPEG file (.jpg)	Hard drive (magnetized sector)	JPEG format	Image
Email file (with embedded attachments	Hard drive (magnetized sector)	Internet Message Format (e.g., RFC 5322 compliant)	Message
USMTF Message file	A specific government network	USMTF Format	Message
Passport	Paper document; (may include photographs, RFID tags)	ID formats, security marking formats …	Name, Personal data, Passport number, Visas …
Title Deed	Official paper document	Varies	Varies
Report	Varies	Varies	Varies
Overlay Sheet (e.g. Map Overlay Sheet – see Figure 2)	Acetate sheet	MIL-STD-2525 Symbols; FM 101-1-5 Operational Terms and Graphics	Map overlay

IX. ATTRIBUTES OF INFORMATION ARTIFACTS

Information artifacts have attributes along a number of distinct dimensions, treated in LLO modules of the IAO. Terms in these modules will be applied to explicate information relating to IAs of different types, and to annotate data pertaining to IA instances with the help of relations mentioned above. Some dimensions of IA attributes are common to all areas, both military and non-military, including: *Purpose*, *Lifecycle Stage* (draft, finished version, revision); *Language*, *Format*, *Provenance*, *Source* (person, organization), and so forth.

Along the dimension of *Purpose* we distinguish:

- Descriptive purpose: scientific paper, newspaper article, after-action report
- Prescriptive purpose: legal code, license, statement of rules of engagement
- Directive purpose (of specifying a plan or method for achieving something): instruction, manual, protocol
- Designative purpose: a registry of members of an organization, a phone book, a database linking proper names of persons with their social security numbers

whereby it should be stressed that one and the same IA may of course serve multiple purposes.

As is shown in Table 3 IAO-Intel will include additional LLOs relating to attributes of importance to the intelligence domain such as: *Classification*, *Encryption Status*, *Encryption Strength*, and so forth. IAO-Intel will also include terms representing specific IA *Purposes* such as: informing the commander, providing targeting support, intelligence preparation of the battlefield.

TABLE 3. DIMENSIONS OF INFORMATION ARTIFACT ATTRIBUTES

Role in the Intelligence Process (JP 3-0, III-11) Priority Intelligence Requirement (PIR) Commander's Critical Information Requirement (CCIR) Essential Element of Information (EEI) Essential Element of Friendly Information (EEFI)	
Confidence Level (JP 2.0, Appendix A) Highly Likely Likely Even Chance	 Unlikely Highly Unlikely
Discipline (JP 2.0, I-5) Legal Ideology Religion Propaganda	 Intelligence Signal Human Rumor intelligence Web intelligence
Intelligence Excellence (JP 2.0, II-6) Anticipatory Timely Accurate Usable	 Complete Relevant Objective Available

Table 3 illustrates fragments of some of the dimensional hierarchies specific to IAO-Intel, with their doctrinal sources.

X. EXAMPLES OF USE OF IAO-INTEL IN ANNOTATION

As should by now be clear, IAO-Intel relates not merely to textual documents but to information artifacts of all types including maps, videos, photographic images, websites, databases, and so forth, both unstructured source documents and official documents of many different varieties. Consider, the Modified Combined Obstacle Overlay (MCOO), taken from JP 2-01.3 [15] and illustrated in Figure 2. (We refer to this as example IA#1 in what follows.) An MCOO is defined as:

> A joint intelligence preparation of the operational environment product used to portray the militarily significant aspects of the operational environment, such as obstacles restricting military movement, key geography, and military objectives.

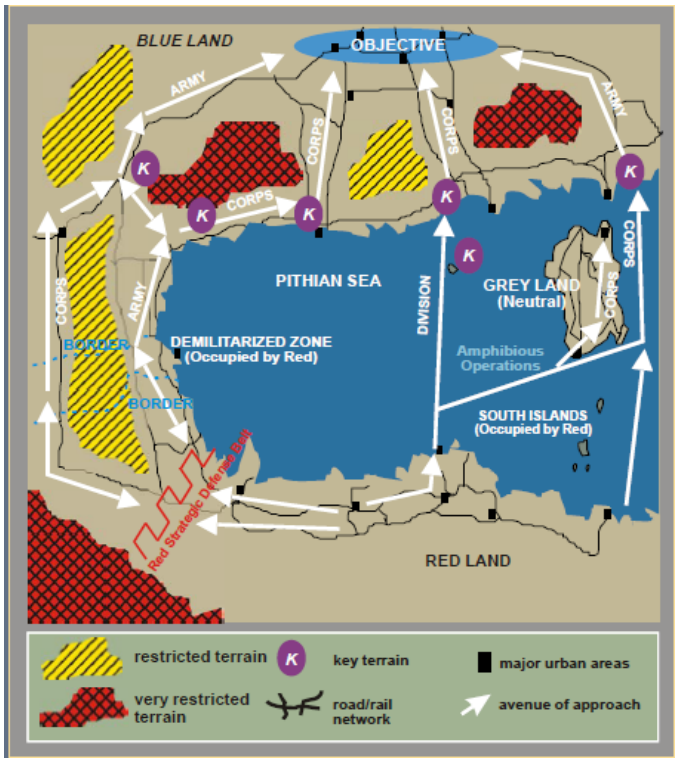


Figure 2: Modified Combined Obstacle Overlay (example IA#1)

We assume that IA#1 has been prepared as part of some given plan, IA#2. Both IAs #1 and #2 will then be referred to in multiple further IAs including multiple databases compiled during planning, execution and outcomes assessment. Relevant terms used in the data models associated with these data models will have been explicated using terms from IAO-Intel. The latter terms can then be used along the lines described in [9] to create annotations to both #1 and #2 on the basis of the fact that they are referred to in the databases in question. The results will include, for example:

a) annotations to the attributes of IA#1*:*

- ICE: MCOO
- IBE: Acetate Sheet
- *uses-symbology* MIL-STD-2525C
- *authored-by* person #4644
- *part-of* plan IA#2

b) annotations relating to the aboutness of IA#1

- Avenue of Approach
- Strategic Defense Belt
- Amphibious Operations
- Objective

and so forth. Used in conjunction with the skill ontology and the person database the annotations above will enable a planner to retrieve (for example) all MCOOs relating to amphibious operations authored by persons with certain skills.

Consider, as a second example, a collection of documents prepared according to FM 6-99.2 [16], for example of types:

Intelligence Report [INTREP]
Intelligence Summary [INTSUM]
Logistics Situation Report [LOGSITREP]
Operations Summary [OPSUM]
Patrol Report [PATROLREP]
Reconnaissance Exploitation Report [RECCEXREP]
SAEDA Report [SAEDAREP]

Suppose further that we need to cross-reference these with comparable sets of documents prepared by other commands, and that we need to do this in such a way as to extract and process the information computationally. FM 6-99.2 provides definitions of the mentioned report types, but does not take the step of formulating these definitions computationally. IAO-Intel addresses this problem by providing a common, algorithmically useful, set of ontology terms that is designed to allow consistent explication of these and related types as they appear in different doctrinal resources. The results can then be used for computer-aided aggregation of the data represented using corresponding IA types, cross-checking of mismatches, and so forth.

XI. The DoD Data Services Environment

We can now return to Directive 8320.02 and address the relevance of the work reported above to its successful implementation. As we saw, the Directive requires that 'all salient metadata be discoverable, searchable, and retrievable' through use of the DoD Data Services Environment (DSE) [6]. DSE's numerous data sources include 35 'supporting taxonomies' derived from pre-existing terminology resources. Problems arise, however, because the latter have been constructed on the basis of multiple distinct methodologies (for example as concerns the formulation of definitions). When, on August 25, 2013, the DSE was queried for information on "location", the DSE reported 660 possibly relevant sources of information. When the DSE was queried for "unit types," 882 possibly relevant sources of information were reported. When types of "ground vehicles" were queried for, 175 possible relevant sources of information were reported. Such redundancies present obstacles to discovery, search and retrieval. They arise because different compilers of authoritative data describe entities of the same types in heterogeneous ways. This thwarts the sort of coherent integration that is required for the mounting of what, in [6], we referred to as the "massing of intelligence fires".

One problem is that while the terms in thesauri and glossaries can be used in annotations, the value derived therefrom is limited above all because they do not allow the benefits of inferencing and of rapid introduction and definition of new terms which are provided by a framework of well-constructed ontologies along the lines described in [10]. There we show how reference ontologies can be quickly expanded with new content to meet emerging data representation needs and in such a way that data annotated with the newly added terms is automatically integrated with existing data.

Imagine, for example that we have two large bodies of data describing (A) chemicals (properties, costs, manufacture, transport, supply, and so forth), and (B) explosives manufacture (raw materials, persons and skills involved, processes and equipment and safety measures used). We will have satisfied Directive 8322.20 in maximizing discoverability if we annotate each body of data in accordance with corresponding term repositories, which we can assume to have been independently developed. Suppose now, however, that we are called upon to integrate the data in (A) with the data in (B). Here these annotations will likely provide no assistance, which will in turn lead to calls for the creation of a third term repository to be used in efforts to annotate the combined (AB) data. The results of these efforts will then once again likely provide no assistance when (AB) data itself needs to be integrated with, say, data about explosives financing.

Where, in contrast, the systems for annotating (A) and (B) reflect a common ontological approach, then new annotation resources for the merged data can be easily be developed by reusing the initially developed ontologies in the formulation of both composite terms and corresponding definitions [10].

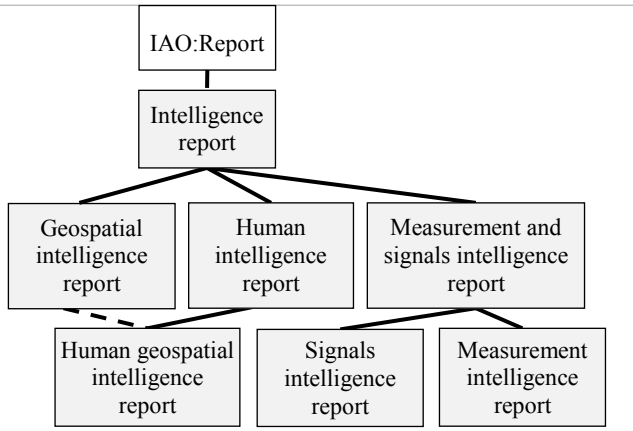
A further problem is that the need to create new terminology resources for the annotation of such merged content may lead to the need for corrections of the initial terminology resources. Such corrections may have expensive consequences: either they will break interoperability with the results of earlier annotation efforts, or – if resources are invested to correct already existing annotations to make them conform to the new usage – they will have unforeseen consequences for third parties who have been relying on the older resources to be maintained consistently through time. Such problems are minimized where terminology resources are developed in tandem from the very start as parts of a single suite of ontology modules developed using common principles, exactly as is proposed by our DSGS-A strategy. We believe that only a strategy of this sort can satisfy the requirement that data, information, and IT services are 'made visible, accessible, understandable, trusted, and interoperable throughout their lifecycles for all authorized users.' [5]

XII. Semantic Technology is Not Enough

The strategy underlying DSE has much in common with a strategy adopted widely in the semantic technology community under the heading of Linked Open Data, a strategy often involving the use of the Dublin Core Metadata Element Set as controlled vocabulary. We believe that the Dublin Core can serve as reliable controlled vocabulary for describing IA data only where the information artifacts in question are themselves artifacts formulated using RDF or some other W3C recommended syntax, and unfortunately this is not the case for many of the artifacts at issue here. We believe further that the Linked Data approaches cannot solve the problems of silo-formulation in the IC for the results outlined already in section XI above. The semantic technology community draws a distinction between two levels of interoperability: Level 1, resting on shared term definitions (for example drawn from the Dublin Core), and Level 2, of what is called Formal Semantic Interoperability. As is recognized at [17], Level 1 is 'so open-ended that it quickly leads to a proliferation of custom-built solutions incompatible with each other, such as

metadata expressed in document formats that require customized software to read and data models that cannot easily be mapped to generic, interoperable representations such as those expressed in RDF.' Level 2 is designed to solve these problems by requiring that all IAs are described via metadata formulated using RDF. Unfortunately RDF (or even OWL) is no panacea. Multiple conflicting ontologies can be formulated in RDF terms, yet still remain conflicting.

The solution, again, must rely on shared development of a single suite of modularized ontologies, in which not only the same formal language is used, but also consistent definitions populating downward from a common upper level such as BFO – and we note in this connection a parallel with the way in which joint doctrine is elaborated, in a process that is designed to ensure (at least ideally) that the same term is defined and used consistently across the 80 plus Joint Publications (JPs) that address the various aspects of joint warfare in accordance with JP 1-02 [2].



The above IAO-Intel terms are defined by using terms from the ontologies below with the help of relations such as *is-about*, *created-by*, *derives-from* and so forth [7].

Figure 3. Top: Terms from IAO (unfilled) and IAO-Intel (grey) ontologies. Taxonomical hierarchies: asserted – solid lines, inferred – dashed lines. Bottom left: Domain ontologies. Bottom Right: IAO-Intel LLOs.

XIII. CONCLUSION

To summarize: IAO-Intel forms part of a collection of ontologies that is being applied primarily to the explication of data models and other terminology resources of importance to DCGS-A. The terms in these ontologies are linked together logically in virtue of the fact that each ontology uses terms which are defined in terms of other ontologies belonging to this same suite (as illustrated in Figure 3). This strategy for ontology development has been tested in use over several years in the domain of biomedical informatics, and is gradually being adopted also in other domains, including for example the domain of modeling and simulation, where the identifying authoritative data sources is needed to ensure realistic scenarios [18]. One principal feature of the strategy is that it provides a standard means for defining new ontologies in light of emerging needs, in a way that guarantees consistency with the ontologies already created and with the data annotated in their terms. We believe that this feature makes the strategy particularly useful in addressing the emerging challenges to the intelligence analyst in accordance with DoD directives concerning discovery, retrieval and search.

ACKNOWLEDGMENTS

Work on IAO-Intel was supported by I2WD. Thanks are due also to Mathias Brochhausen, Werner Ceusters, Mélanie Courtot, Janna Hastings, James Malone, Bjoern Peters, Jonathan Rees, and Alan Ruttenberg for their work on IAO.

REFERENCES

[1] Friedrich Wilhelm von Steuben, Regulations for the order and discipline of the troops of the United States, 1792, http://x.co/1dJEk.

[2] Department of Defense Dictionary of Military and Associated Terms, 2013, http://www.dtic.mil/doctrine/new_pubs/jp1_02.pdf.

[3] Leo Obrst, Patrick Cassidy, "The need for ontologies: Bridging the barriers of terminology and data structure", Geological Society of America Special Paper 482, 2011.

[4] Leo Obrst, Terry Janssen, Werner Ceusters (eds.), Ontologies and Semantic Technologies for the Intelligence Community. Amsterdam: IOS Press, 2010.

[5] Sharing Data, Information, and Information Technology (IT) Services in the Department of Defense, DoD Instruction 8320.02, August 5, 2013, http://www.dtic.mil/whs/directives/corres/pdf/832002p.pdf.

[6] DSE Data Services Environment, https://metadata.ces.mil/dse.

[7] https://code.google.com/p/information-artifact-ontology.

[8] David Salmen, Tatiana Malyuta, Alan Hansen, Shaun Cronen, Barry Smith, "Integration of intelligence data through Semantic Enhancement", Proceedings of the Conference on Semantic Technology in Intelligence, Defense and Security (STIDS), 2011, CEUR 808, pp. 6–13.

[9] Barry Smith, Tatiana Malyuta, David Salmen, William Mandrick, Kesny Parent, Shouvik Bardhan, Jamie Johnson, "Ontology for the Intelligence Analyst", CrossTalk: The Journal of Defense Software Engineering, November/December 2012, pp. 18–25.

[10] Barry Smith, Tatiana Malyuta, William S. Mandrick, Chia Fu, Kesny Parent, Milan Patel, "Horizontal integration of warfighter intelligence data. A shared semantic resource for the Intelligence Community", Proceedings of STIDS Conference, 2012 (CEUR 996), pp. 112–119.

[11] Ron Rudnicki, Werner Ceusters, Shahid Manzoor and Barry Smith, "What particulars are referred to in EHR data?", American Medical Informatics Association 2007 Annual Symposium, 2007, pp. 630–634.

[12] Ron Rudnicki, "DCGS-A Ontology Program Explication Procedures", MS, 2013.

[13] Barry Smith and Werner Ceusters, "Ontological Realism as a methodology for coordinated evolution of scientific ontologies", Applied Ontology, 5 (2010), pp. 139–188.

[14] Basic Formal Ontology 2.0, http://ontology.buffalo.edu/BFO/Reference.

[15] Joint Publication 2-01.3 Joint Intelligence Preparation of the Operational Environment, 16 June 2009.

[16] U.S. Army Report and Message Formats (FM 6-99.2), April 2007, http://armypubs.army.mil/doctrine/DR_pubs/dr_a/pdf/fm6_99x2.pdf.

[17] Dublin Core User Guide, Last modified September 6, 2011, http://wiki.dublincore.org/index.php/User_Guide.

[18] Saikouy Diallo, Jose Padilla, "Military Interoperability Challenges", Handbook on Real-World Applications in Modeling and Simulation, Wiley, 2012, pp. 298–332.

Managing Semantic Big Data for Intelligence

Anne-Claire Boury-Brisset
Defence Research and Development Canada
Québec, Canada
anne-claire.boury-brisset@drdc-rddc.gc.ca

***Abstract*— All-source intelligence production involves the collection and analysis of intelligence data provided in various formats (raw data from sensors, imagery, text-based from human reports, etc.) and distributed across heterogeneous data stores. The advance in sensing technologies, the acquisition of new sensors, and use of mobile devices result in the production of an overwhelming amount of sensed data, that augment the challenges to transform these raw data into useful, actionable intelligence in a timely manner. Leveraging recent advances in data integration, Semantic Web and Big Data technologies, we are adapting key concepts of unified dataspaces and semantic enrichment for the design and implementation of a R&D intelligence data integration platform MIDIS (Multi-Intelligence Data Integration Services). The development of this scalable data integration platform rests on the layered dataspace approach, makes use of recent Big Data technologies and leverages ontological models, and semantic-based analysis services developed for various purposes as part of the semantic layer.**

Keywords—intelligence, data integration, knowledge extraction, ontology, Big Data

I. INTRODUCTION

The advance in sensing technologies, the acquisition of new sensors, and use of mobile devices result in the production of an overwhelming amount of sensed data, that augment the challenges to transform these raw data into useful, actionable intelligence in a timely manner. Consequently, intelligence operators and analysts have to deal with ever-increasing amounts of ISR data and information from various sources (SIGINT, IMINT, GEOINT, HUMINT, OSINT, etc.), produced in disparate multiple media formats (raw data sets from sensors, e.g., video, images, sound files, as well as human reports and open source text), and distributed across different systems and data stores.

As part of a research project conducted within the Intelligence and Information Section at Defence Research and development Canada (DRDC) – Valcartier, we are investigating advanced concepts, techniques and technologies in order to provide enhanced capabilities for the management and integration of large-scale heterogeneous information sources and intelligence products made available to intelligence operators and officers in support of the production of intelligence and sense-making activities.

Our ultimate goals are to:

- Provide timely and relevant information to the analyst through intuitive search and discovery mechanisms;
- Provide a framework facilitating the integration of heterogeneous unstructured and structured data, enabling Hard/Soft fusion and preparing for various analytics exploitation.

This paper describes ongoing research for the design and implementation of a prototype for scalable Multi-Intelligence Data Integration Services (MIDIS) in support of these objectives, based on a flexible data integration approach, making use of Semantic Web and Big Data technologies. The paper is organized as follows. In the next section, we present recent work addressing multi-intelligence data integration, followed by a short introduction to Big Data challenges. Section IV describes the proposed architecture for large-scale intelligence data integration and analysis and details the main components of the resulting architecture. Section V provides details about the implementation using Big Data technologies. Section VI provides some conclusions and directions for future work.

II. MULTI-INTELLIGENCE DATA INTEGRATION

Intelligence is about data management and processing: 1) data collection from various sources, 2) data analysis for the production of intelligence and 3) dissemination of intelligence products. Intelligence data management nowadays presents the following characteristics:

- Increase of sensor data volume (terabytes to exabytes);
- Heterogeneity: multiple data formats and standards, mix of structured and unstructured;
- Need to quickly acquire and process intelligence information;
- Agility is required to be able to incorporate new data sources;
- Support to data exploitation: each piece of data represents some part of a situation, intelligence data contain entities that must be understood and correlated.

Data integration aims at combining data that reside at distributed, autonomous, and heterogeneous data sources into a

single consistent view of the data [7]. Traditional approaches propose either centralized or federated data integration. The centralized approach requires heavy pre-processing through extract, transform load (ETL) processes while the latter can denote performance and complex transformations issues. These approaches have been largely detailed and challenged in the literature, and they have been recently exposed by Singleton [19] as part of a research work in the military domain.

As an alternative to these approaches to cope with large-scale heterogeneous data management, Franklin, Halevy and colleagues [11] proposed the concept of dataspaces as a new abstraction for information management. That is, it promotes a flexible co-existence approach for the incorporation of heterogeneous data into a dataspace, and a description of the concepts of the domain at a higher-level of abstraction. Integration in terms of schema harmonization is realized in a pay-as-you-go approach [12].

Looking for a flexible data integration solution to deal with the ever increasing heterogeneous data sources in the intelligence domain and information fusion, S. Yoakum-Stover proposed a framework to implement this scheme [20, 21]. Based on that approach, D. Salmen and colleagues [16] described their implementation of the approach. It rests on the definition of a data integration framework (DRIF), also called Data Description Framework (DDF) in previous papers, based on a unified data integration model. The idea is to define a simple data representation scheme to encapsulate every piece of data from heterogeneous sources into a unified representation. The elementary constructs are composed of signs, terms, concepts, predicates and statements, the latter being conceptually similar to the Semantic Web Resource Description Framework (RDF) triple composed of subject, predicate, and object.

Based on this unified scheme, the dataspace is organized into several layers, namely:

- Segment 0 contains the external data sources and systems from which relevant data are extracted;
- Segment 1 (unstructured data) represents the data store for artefacts;
- Segment 2 (structured data) is the universal store for data structured according to the unified representation scheme;
- Segment 3 (data models) contains the representation of data models and ontologies to facilitate the mapping and integration of heterogeneous data.

The concepts underlying the unified dataspace have been implemented as part of the US Army's Distributed Common Ground System (DCGS-A) Cloud initiative [17]. Moreover, to address semantic heterogeneity, B. Smith and colleagues [17] propose a strategy for the integration of diverse data through semantic enhancement, by adding a semantic layer to the data (explicitly represented in segment 3).

Leveraging this approach, we are adapting the underlying concepts for the design and implementation of a R&D intelligence data integration platform MIDIS (Multi-Intelligence Data Integration Services) to meet our requirements in support of intelligence. In previous research, our team has developed several intelligence support tools in support of collation and intelligence production, and knowledge-based systems on top of military domain ontologies to meet various analysis requirements [15]. Some of the relevant components from these tools have been incorporated as services as part of a SOA-based Intelligence Science and Technology Integration platform (ISTIP) in development.

The data access component had to be further developed in this platform to provide the ability to dynamically ingest, integrate and manage data from various intelligence sources. Consequently, MIDIS aims at enriching the data access component of the ISTIP platform to provide the set of services needed to ingest multiple intelligence data formats available, transform them into a unified model, and make these data accessible, searchable and exploitable (e.g. data mining) in support of intelligence analysis.

The design and development of MIDIS as a scalable data integration platform rests on the layered dataspace approach and makes use of Big Data technologies. Moreover, we leverage ontological models, and semantic-based analysis services developed for various purposes as part of the semantic layer within the architecture described in section IV.

III. BIG DATA CHALLENGES

Considering the huge amount of data produced every day in both the commercial and the defense areas, the Big Data paradigm promotes novel approaches and technologies for data capture, storage and analytics to deal with "massive volume of unstructured and structured that cannot be managed and processed with traditional databases and software approaches" [3].

Big Data are initially characterized according to 3 Vs, namely: 1) Volume or scalability: ability to manage increasing volumes of data, for storage and analysis; 2) Variety: heterogeneity of data types, data formats, semantic interpretation; 3) Velocity: timeliness or rate at which the data arrives and time in which it must be acted upon. Additional Vs are sometimes added, to denote the Veracity of data, as well as the Value that can be extracted from Big Data.

The problem of information overload is not new, but it is amplified in the new information era. Big Data challenges encompass most data management processes, i.e. data capture, curation, storage, search, sharing, analysis, and visualization. In our research work, we are interested by Big Data solutions for on-the-fly integration of heterogeneous data from various sources, effective search among heterogeneous possibly inconsistent data sets, while managing data granularity and consistency. Some of these will be discussed later in the paper.

IV. ARCHITECTURE COMPONENTS

The implementation of the unified dataspace approach points toward Big Data technological solutions, as they provide scalability, elasticity, replication, fault-tolerance, and parallel processing. Next, we present the proposed global architecture

for intelligence data integration, its main components (data ingestion process, ontology support and semantic enrichment, search and analytics), and interactions with other reasoning modules.

A. *Global architecture: from Collection to Analysis*

Figure 1 represents the high-level architecture and data flow, from data collected from heterogeneous data stores, their ingestion into the dataspace, to intelligence analysis by specialized reasoning services. The key components include:

- Data ingestion from heterogeneous sources formats and integration into the unified dataspace segments;
- Ontology-based semantic enrichment;
- Data querying and analytics;
- Interactions with external reasoning modules.

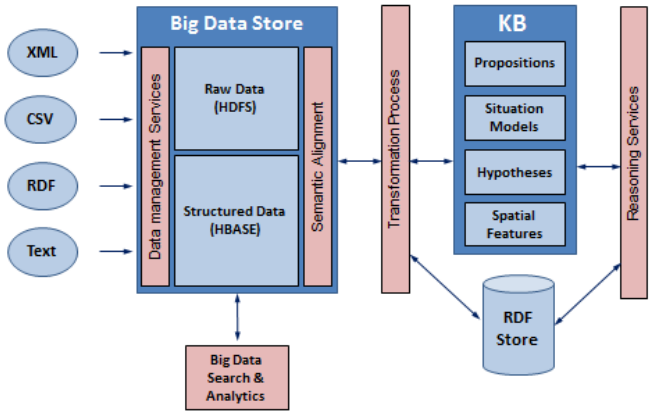


Figure 1: Intelligence data integration and analysis framework

B. *Data ingestion*

The system ingests intelligence data from representative sources provided in heterogeneous formats, in order to illustrate the integration of a variety of intelligence data as used by intelligence analysts to conduct multi-intelligence all-source analysis. A subset of the considered data sources in this context include:

- Structured data coming from intelligence or operational database, including track data;
- Intelligence reports;
- Imagery database;
- Data from a Content Management System;
- Internet open source (e.g. Twitter).

The data ingestion pipeline is applied to structured and unstructured data (cf. Fig. 2) as follows. Figure 2 illustrates the data flow and transformation process from external data sources, and shows explicitly how data pieces move to different segments.

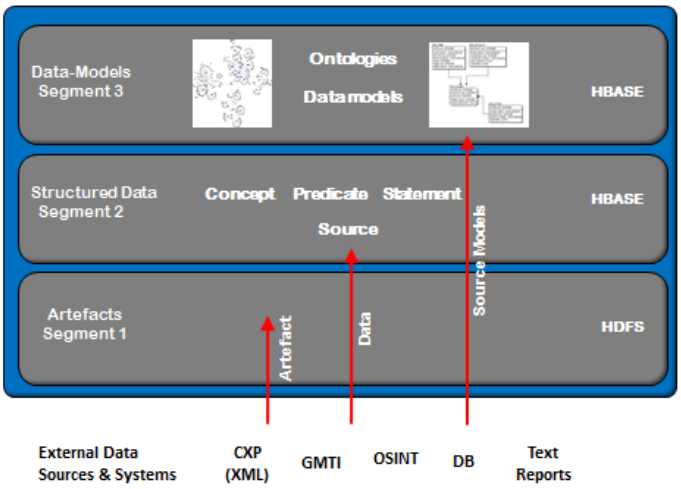


Figure 2: UDS layered architecture and data flow (adapted from Yoakum-Stover, 2012 [22])

1) *Structured data*

The ingestion pipeline for structured data processes various structured data sources (RDB, CSV, XML, RDF format) in order to populate the UDS in segments 1, 2, 3. The approach makes use of a XML configuration file generic enough to process each data schema provided (e.g. WSDL web service provides the XML schema to be processed). Data files are then parsed to extract data of interest and load them according to UDS constructs, i.e. concepts, predicates, statements into the UDS and reference to the source in segment 2, source model in segment 3, while the imported data source is ingested in segment 1.

2) *Unstructured data: annotation and extraction*

Unstructured data (e.g. intelligence reports, documents) is processed according to a text analysis pipeline using semantic annotation and knowledge extraction services supported by domain ontologies. Documents are analyzed and semantically annotated using concepts instances from domain ontologies (named entities, people, location, …). Then, knowledge in the form of statements (e.g. X is_located_at Loc) is extracted using pattern matching rules. These processes use the popular GATE platform (General Architecture for Text Engineering) [4] as the underlying natural language processing component. Documents and their annotations are stored in the segment 1 while extracted facts and metadata that provide meta-information about the documents are stored in the segment 2 according to the unified model (structured data). Metadata of interest include data provenance, uncertainty, temporal and spatial information.

In military intelligence context, imagery data sources (images, videos) are currently managed using metadata according to standard agreements (e.g. Stanag 4559) to facilitate information sharing (e.g. coalition operations). The next step in our architecture will be to adapt and enrich the data ingestion process for this type of source, possibly including automated information extraction.

C. Ontology and semantic enrichment

The proposed integration approach rests on the exploitation of domain ontologies to facilitate the harmonization of data models in a flexible and incremental manner.

1) Ontology engineering

Ontologies describe flexible and extensible conceptual models that explicitly represent the concepts in a domain of interest and the relationships that exist between them. Ontologies have been considered as an enabler for information integration and have also been exploited in support of information management or reasoning to meet different needs:

- To provide a standardized vocabulary and a taxonomy of the concepts in the domain of interest and facilitate information sharing;
- To support text analysis and semantic annotation;
- To perform federated semantic searches;
- To perform automated reasoning on top of the ontology and business rules;
- As a knowledge base (instances, relationships) to capture information about the domain.

In the military domain, ontologies have been developed for the last decade to meet various requirements: ontologies in support of command and control [13], low-level and high-level information fusion, in particular situation and threat assessment [1], or intelligence analysis [18, 2].

At DRDC, domain ontologies have been developed and exploited in order to fulfill command and control as well as intelligence requirements in different specific application contexts, namely:

- Maritime domain ontology in support of threat analysis and anomaly detection.
- Situation awareness ontologies to support knowledge management and knowledge mapping applications.
- Ontologies related to terrorism and Improvised Explosives Devices (IED) for ontology-based semantic annotation of texts in support of intelligence collation.

In the evolving military context, such as counterinsurgency and counter-terrorism, cyber-warfare, civil-military operations, the human terrain is a key component. The National Geospatial intelligence Agency (NGA) has undertaken the development of human geography data standards and models that define top-level constructs and a set of sub-models encompassing topics of interest such as religion, language, demographics, ethnicity, groups, culture among others. The key high-level concepts are composed of *Feature* to represent temporally persistent real-world phenomenon, *Event* to represent instantaneous or short-duration real-world phenomenon, *Actor* to represent an intentional entity that acts or has the capability of acting as a participant in an event (individuals, groups), and *Information* to collect non-geometric properties of other entity types. Based on these models, we have developed an ontology of human geography to formally represent the entities present in these models, thus enabling automated reasoning upon it. These models provide knowledge to support applications such as the Intelligence Preparation of the Operational Environment, terrain analysis, and social network analysis that require a formal representation of the human terrain elements.

In some of our previously developed ontological models, concepts are derived from the hierarchy structure of the JC3IEDM (Joint Command and Control Information Exchange Data model) and its subsequent MIP Information Model revisited and represented as a UML model. The model decomposes battlespace entities along *Objects* and *Action/Event* high-level concepts. Consequently, key high-level concepts contained in such ontologies comprise: individuals, groups and organizations, events that occur and activities that are conducted in the area of operations, their location, the characterization of the reported information, etc. Ontologies also formally represent the relationships that may exist between these entities. Of course, the spatio-temporal dimension inevitably associated to these concepts has to be modelled accordingly.

Domain ontologies are developed incrementally by adapting recognized multi-stages development methodologies, leveraging as much as possible military models and doctrine documents. Such development approaches promote a modular, layered approach to ontology construction, built on top of foundational or upper-ontologies (e.g. SUMO, BFO, Dolce, etc.) that represent generic concepts, which can be further extended to represent more specific concepts in the domain of interest according to a hierarchical taxonomic structure.

In the intelligence domain, the set of concepts of interest is derived from a thorough analysis of key processes and data sources, e.g. collation and analysis phases, in order to capture the essential entities in the ontological model. While elements of such knowledge are captured in some existing models, it is of interest to develop the corresponding ontological models and integrate them on top of some upper-level ontologies. Looking at the high-level concepts taxonomy of our ontologies, and some existing upper ontologies mentioned above, they present similarities in the high-level decomposition. BFO (Basic Formal Ontology) [13, 18] as well as the UCore Semantic Layer are models that we are leveraging to benefit from prior modeling efforts. We are revisiting and integrating them as part of this work.

Moreover, domain ontologies are being extended as new data sources or applications required additional concepts to be considered, and as the domain evolves (e.g. human terrain, cyber). As mentioned in [18], rigorous management and governance principles have to be applied to ensure consistency and non-redundancy.

Domain ontologies are developed using the OWL language based on Description Logic due to its popularity, interoperability facilitating the reuse of ontology parts, expressiveness and tractability to represent domain knowledge with expressive semantics. Consistency checking tools are used to ensure that the developed ontologies are free of inconsistencies.

2) Semantic Enrichment

Semantic Enrichment (SE) [17] is a process for horizontal data integration based on the use of ontologies to integrate and semantically enhance data models. The enhancement is accomplished by annotating (tagging) the models by the terms of the ontology(ies), thus linking together the various resources in a semantically coherent way.

According to the layered organizational structure of data in the unified dataspace, the suite of ontologies and source data models are part of segment 3. Mappings between terms of the ontologies and labels in the data models are explicitly defined at this level too, so that data models are harmonized using the semantic layer.

Consequently, using this extra semantic layer, additional semantic power (inferencing) can be exploited by query engines, or reasoners (e.g. exploiting "same_as" relations between terms linked by the same concept in the reference ontology).

To fulfill semantic enrichment approach consisting of semantically linking data, unstructured documents are also processed by exploiting the terms and structure of ontologies.

D. Data search and analytics

As mentioned above, this work leverages and extends previous research we have conducted in support of intelligence, e.g. the provision of information management and exploitation services to support the analyst in his activities: semantic search engines, filtering, notification/alert services, etc.

The focus in the present research is to provide scalable solutions for large-scale data management and analysis. Consequently, we are investigating various techniques and solutions that fulfill analysts' increasing needs in terms of:

- Analytics from large data sets: data mining, data/document clustering, data correlation among various data sets, etc.
- Efficient search and retrieval within unstructured and structured data sets.

Multi-intelligence data are ingested into the dataspace segments 1 and 2 as presented above. Consequently, efficient indexing and search techniques and tools have to be proposed both for data in segments 1 (unstructured world) and in segment 2 (structured data). While analytics tools benefit from Big Data technologies (batch distributed processing), the required search tools have to provide real-time performance results. Some techniques are discussed in section V.

E. Interface with intelligence reasoning modules

While MIDIS first aims at integrating intelligence data from heterogeneous data sources for further retrieval and exploitation, it is part of a comprehensive architecture (ISTIP) for the analysis and production of intelligence. Thus, interfaces to facilitate data flow/transformation between the UDS and reasoning components are required (cf. Fig. 1). Consequently, we provide mechanisms and services to export data through a transformation process into appropriate formats to/from existing intelligence analysis modules.

- Intelligence reasoning services make use of various rich data formats required as input by their engine (e.g. rule-based reasoning and/or case-based reasoning), e.g. propositions, situation model, spatial feature, hypotheses structures.
- Data can also be exported as RDF into a graph representation to be used by various reasoning services, e.g. social network analysis algorithms.

Inversely, data produced by the various reasoning modules can be persisted in the dataspace. They are ingested back as new data in the UDS through the appropriate transformation process, thus made discoverable for subsequent processing.

V. TECHNOLOGICAL ASPECTS

The implementation of our multi-intelligence data integration system leverages emerging Big Data and SOA technologies.

A. Big Data Technologies

To cope with the processing of ultra-large scale data sets, Big Data technologies exploit distributed storage and processing. The open source Apache Hadoop Framework [5] allows for the distributed processing of large data sets across clusters of computers using simple programming models. It provides several components, including the MapReduce distributed data-processing model, Hadoop Distributed File System (HDFS), and HBase [6] distributed table store. These main components and emerging tools are being exploited for the implementation of our integration architecture (Cloudera's platform).

1) Data ingestion

Data ingestion benefits from Hadoop MapReduce distributed processing for large data sets. As presented above, structured data ingestion is done by using a XML configuration file for each data format. Data files are then parsed via MapReduce and loaded into the UDS.

Artefacts data are stored in HDFS in segment 1, structured data are stored in HBase in segment 2, and data models in segment 3 in HBase as well.

Knowledge extraction from textual documents using semantic text analysis services were not initially implemented

using parallel processing. We are considering their adaptation into Hadoop environment to benefit from distributed processing of large documents corpus and are also looking at alternate approaches such as those proposed in Lin and colleagues' book [8]. Additional envisioned services for extraction value from textual intelligence reports datasets include cross-document co-referencing in HDFS.

2) Indexing / Query

For users (or services) to retrieve relevant information from the HBase UDS in near real-time, we aim at providing efficient indexing and query solutions.

First, considering out of the box query tools, the Hive query engine has demonstrated poor performance. The recent Cloudera Impala query engine is being experimented, the performance is improved due to the fact that it supports direct query on HBase indexes and does not use MapReduce.

Moreover, several input data formats to the UDS will be as RDF triples (metadata extracted from text, imagery data tagging, data extracted from content management systems, etc.). Conceptually, the UDS segment 2 can be considered as a HBase quad store where the fourth element added to the triple refers to the source (named graph). We are looking at techniques to perform efficient queries to retrieve RDF data in this context (e.g. extraction of graphs for Social network analysis).

One interesting approach is provided by Rya [14] that introduces storage methods, indexing schemes, and query processing techniques that scale to billions of RDF triples across multiple nodes, while providing fast and easy access to the data through conventional query mechanisms such as SPARQL. Rya proposes a method of storing triples by indexing triples across three different tables corresponding to the permutations of triple patterns, i.e. (Subject, Predicate, Object), (Predicate, Object, Subject), and (Object, Subject, Predicate). We are experimenting with this approach, and are exploiting OpenRDF Sesame (SPARQL) for HBase [10].

Preliminary tests are being done with various data sources, as well as using the LUBM benchmark dataset [9] to assess the performance and compare with other approaches.

3) Analytics

While intelligence analysis requires specialized reasoning tools and human intervention, Big Data Analytics may reveal interesting insights from the analysis of large data, (e.g. predictive/trend analysis) by using appropriate techniques such as data mining. Apache Mahout is one of the first distributed machine-learning open source framework built on top of Hadoop. It is a candidate for data clustering, classification, collaborative filtering, recommendation, or profiling that we are considering in order to demonstrate value-added from data using Big data analytics.

B. SOA

Service Oriented Architecture (SOA) has emerged as the predominant paradigm for the building of flexible and scalable architectures in net-centric environments. SOA is an architectural discipline that relies upon the exposure of a collection of loosely-coupled, distributed services which communicate and interoperate via agreed standards across the network. Some benefits are directly based on the principles of service orientation, mainly: services are loosely coupled, autonomous, discoverable, composable and reusable. Consequently, SOA principles offer an appropriate approach to data integration. The services can be composed into higher-level applications to support agile business processes. By augmenting the data services layer, and incorporating integration services as described above, the data integration environment will facilitate access to data and discovery, integration of data from diverse sources, and handling of large volume of data.

The envisioned set of services complements the SOA-based Intelligence Science and Technology Integration platform (ISTIP) in development at DRDC Valcartier. This platform already incorporates a set of data representation schemes and relevant services in support of various intelligence analysis tasks and sense-making activities: the analysis of textual documents, (semantic annotation of text based on domain ontologies, and automated extraction of facts from documents based on pattern matching rules), as well as multiple reasoners (rule-based reasoner, case-based reasoner, multiple hypotheses situation analysis) [15]. Our contribution will augment the platform with additional intelligence data services, using flexible and efficient representation schemes. This will facilitate the linking of data among the various sources, in order to make sense of the large amount of data made available to analysts, and provide improved situational awareness.

VI. CONCLUSIONS

In this paper, we have presented the ongoing work that we are conducting for the development of a scalable and flexible intelligence data integration and analysis platform. As part of this initiative, we leverage our previous R&D work using semantic technologies, in particular the suite of ontologies and services that are part of our ISTIP platform. Moreover, we are leveraging a proposed integration approach [22] and adapting it to our needs. We are currently developing data integration components by experimenting with recent Big Data technologies to address scalability and performance.

Big Data technologies represent a shift in terms of programming approach, and their promise produce an increasing interest within the data/information management community. But proposed solutions are still immature, and first experimentations show that they require incremental development and testing stages to improve performance. In our military intelligence context, Big Data performance is critical if these technologies are be used in tactical environments.

While we aim at providing a comprehensive data management and exploitation platform, further research is required to deal with entity resolution, disambiguation, data

cleaning, etc. in this context. Recent research proposed in the Big Data world should provide relevant insight.

A data integration platform can be viewed as a prerequisite to multi-sources information fusion. Work within the hard/soft information fusion community addresses similar challenges, and we looked at them from an architecture perspective. The management of data uncertainty should be considered beyond simple metadata when integrating intelligence data from heterogeneous sources.

We are also investigating approaches to the integration and exploitation of internet open sources in support of intelligence analysis, in particular from social media (e.g. twitter).

REFERENCES

[1] Boury-Brisset, A.-C. Ontology-based approach for Information Fusion, in Proceedings of the 6th International Conference on Information Fusion, Cairns, 8-11 July, Australia, pp. 522-529, 2003.

[2] V. Dragos, Developing a core ontology to improve military intelligence analysis, in International Journal of Knowledge-based and Intelligence Engineering Systems, 17, pp.29-36, IOS Press, 2013.

[3] Gartner, Hype cycle for Big Data, Gartner research report, 2010, also published in 2012.

[4] GATE, General Architecture for Text Engineering, http://gate.ac.uk/index.html.

[5] Hadoop. http://hadoop.apache.org/.

[6] HBase. http://hbase.apache.org/.

[7] M. Lenzerini, Data integration from a theoretical perspective, In Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, 2002.

[8] Jimmy Lin and Chris Dyer. Data-Intensive Text Processing with Mapreduce. Morgan and Claypool Publishers, 2013.

[9] Guo, Yuanbo, Pan, Zhengxiang and Heflin, Jeff . LUBM: A Benchmark for OWL Knowledge Base Systems. Web Semantics. 3(2) July 2005.

[10] OpenRDF. http://www.openrdf.org/.

[11] M. Franklin, A. Halevy, and D. Maier. From databases to dataspaces: A new abstraction for information management. SIGMOD Record, 34(4):27-33, December 2005.

[12] S. Jeffery, M. Franklin, and A. Halevy. Pay-as-you-go user feedback in dataspace systems. In Proc. of SIGMOD, 2008.

[13] B. Mandrick, Creating an extensible command and control ontology, in Int. Journal of Intelligent Defence Support Systems, Vol. 4, No. 3, 2011.

[14] R. Punnoose, A. Crainiceanu, and D. Rapp. Rya: a scalable RDF triple store for the clouds. In Proceedings of the 1st International Workshop on Cloud Intelligence (Cloud-I '12). ACM, New York, NY, USA, 2012.

[15] Roy, J. and Auger, A., The Multi-Intelligence Tools Suite - Supporting Research and Development in Information and Knowledge Exploitation, 16th International Command and Control Research and Technology Symposium (ICCRTS), "Collective C2 in Multinational Civil-Military Operations", Québec City, Canada, June 21-23, 2011.

[16] Salmen D., Malyuta T., Hansen A., Cronen S., and Smith B., Integration of Intelligence Data through Semantic Enhancement, in Proceedings of the 6th international conference on Semantic Technology for Intelligence, Defense, and Security (STIDS 2011), George Mason University, Fairfax, Virginia, November 2011.

[17] B. Smith, T. Malyuta, W. S. Mandrick, C. Fu, K. Parent, M. Patel, Horizontal Integration of Warfighter Intelligence Data. A Shared Semantic Resource for the Intelligence Community, Proceedings of the Conference on Semantic Technology in Intelligence, Defense and Security (STIDS), George Mason University, Fairfax, VA, October 23-25, 2012.

[18] B. Smith, T. Malyuta, D. Salmen, W. Mandrick, K. Parent, S. Bardhan, J. Johnson, Ontology for the Intelligence Analyst, CrossTalk: The Journal of Defense Software Engineering, November/December 2012,18-25.

[19] J. Singleton, Data integration: charting a path forward to 2035, Air War college, research report, Feb. 2011.

[20] S. Yoakum-Stover and T. Malyuta, "Unified data integration for Situation Management," IEEE MILCOM 2008.

[21] Yoakum-Stover S., Malyuta T., Antunes N., A Data Integration Framework with Full Spectrum Fusion Capabilities, Sensor and Information Fusion Symposium, Las Vegas, 2009

[22] S. Yoakum-Stover, A. Eick, Breaking the Data Barriers, DGI, London, 2012.

Context as a Cognitive Process: An Integrative Framework for Supporting Decision Making

Wayne Zachary, Ph.D., Andrew Rosoff
CHI Systems, Inc.
2250 Hickory Road, Suite 150
Plymouth Meeting, PA, 19462, USA
Wzachary@chisystems.com ARosoff@chisystems.com

Lynn Miller, Ph.D., Stephen Read, Ph.D.
University of Southern California
Los Angeles, CA, 90089, USA
Read@usc.edu lmiller@usc.edu

***Abstract*— Multiple lines of research in cognitive science have brought insight on the role that internal (cognitive) representations of situational context play in framing decision making and in differentiating expert versus novice decision performance. However, no single framework has emerged to integrate these lines of research, particularly the views from narrative reasoning research and those from situation awareness and recognition-primed decision research. The integrative framework presented here focuses on the cognitive processes involved in developing and maintaining context understanding, rather than on the content of the context representation at any given moment. The Narratively-Integrated Multilevel (NIM) framework views context development as an on-going and self-organizing process in which a set of knowledge elements, rooted in individual experience and expertise, construct and maintain a declarative, hierarchical representation of the situational context. The context representation that arises from this process is then shown to be the central point of both situational interpretation and decision-making processes at multiple levels, from achieving specific local goals to pursuing broad motives in a domain or theater of action.**

Keywords— situational awareness; recognition-primed decision making; narrative reasoning; self-organizing architecture; decision support systems

I. INTRODUCTION

The current scientific understanding of the role of context in decision-making has evolved in multiple steps over the last forty years. Cognitive science research has long shown that while human actions and decisions are based on the person's environmental context, the decision-making process relies on an internal (cognitive) representation of the context, not directly on the context as sensed (see [3] for a succinct review of this literature]. In the 1980s, convergent research on:

- the study of decision making in its naturalistic setting rather than in laboratory experiments [12,13];
- cognitive skill acquisition theory [31,34]; and
- mental models in cognition, e.g.,[36]

found that the content and organization of an internal representation of the problem instance differentiated the performance of skilled decision makers (DMs) from less-skilled ones. Specifically, these separate lines of research pointed to the fact that expert DMs – across domains – use internal representations of the problem instance in its environmental setting that are richer and more stylized, incorporate multiple levels of abstraction, and take on a structure that enables rapid retrieval of relevant decision-making heuristics and procedures. This latter feature became widely known as recognition-primed decision-making or RPD [14].

In the 1990s, research on the structure of mental models of context across domains began to suggest that there is consistent, hierarchical structure to (expert) mental models. In particular, the work of Endlsey [5,6] developed a theory of the general structure of expert-level context mental models across dynamic, real-time domains. Terming the understanding of the changing external context as Situation Awareness (SA), Endsley identified three increasingly abstract levels:

1. *Perception*, in which the person perceives the status, attributes, and dynamics of relevant elements in the situation and their current states,
2. *Comprehension*, in which the person understands how the perceived elements can impact situational goals; and,
3. *Projection*, in which the person can project the future actions of the elements in the environment forward in time.

There is an explicitly constructive assumption about these levels, in that level 1 information is represented from information directly perceived from the environment, level 2 information is constructed mentally from level 1, and level 3 information is mentally constructed from Level 1 and Level 2 information. SA and RPD theory have led to the development of various decision support applications [9,11,18,20].

While this thread of cognitive research was building an understanding of the role of context from the bottom-up (i.e., building from fundamental insights on human information processing mechanisms), a separate thread of 'top-down' cognitive research unfolded from the 1980s forward. This thread explored how people understand and reason about

Parts of this research were supported by Contract F33615-02-C-6033 from the US Navy. The remainder was sponsored by CHI Systems, Inc. internal research and development.

sequences of action and interaction in which the main source of variability is human behavior. (This aspect is particularly germane to military decision-making, in that it typically involves situations with both adversaries and non-combatants). This research focused on narrative reasoning processes in which the observer/participant constructs, analyzes, and explains complex situations through a narrative (story-telling) process. Specifically, it found that people almost universally use story narratives to represent, reason about, and make sense of contexts involving multiple interacting agents, using (general) motivations and (local) goals to explain both observed and possible future actions. In other words, people were found to generally make sense of their human contexts by either integrating them into a novel narrative or, more commonly, by recounting them as an instance of a commonly-known or culturally based narrative [4,10,26,28]. There is also evidence that people maintain narrative structures mentally and use them to identify, assess, and select behavioral options – that is, to support decision-making [27,28]. These ideas have been widely applied, for example in criminal investigations [1], legal decision-making [21,22], policy analysis and formation [37], and in social interactions [17].

Despite their convergent directions, the bottom-up SA/RPD theories and the top down narrative reasoning theories have not yet met. This paper presents a framework in which such an integration can occur, and explores its benefits for decision support and human-machine integration.

II. Context as Integrated Process

This failure of the two theories to integrate immediately points out several unmet challenges for decision support. For example, changing patterns within SA do not, by themselves, present the DM with any easy way to see alternative narrative interpretations for the context dynamics (making DMs more vulnerable to deception). SA theory and RPD theory have worked best in contexts that involve well-defined problem-solving in bounded problem domains, such as putting out fires [15], piloting aircraft [7], and controlling complex mechanical systems [8]. Even though they have successfully been automated as cognitive models and used for training and advisory purposes, the upper levels of context in SA theory do not yet articulate with the narrative level of context representation (and the reasoning processes associated with that level). At the same time, decisions made at a narrative level are not easily instantiated into action specifics without direct access to the more detailed understanding of situational details available at the lower levels of the framework. For this reason, narrative reasoning has proven most useful in applications that involve non-real-time sense-making (e.g., [1, 21,22]).

The authors and colleagues have conducted a line of research to develop and apply computational models of expert cognition in various domains, both to test and refine cognitive theory and to develop support for decision making and decision training. That research initially focused on operationalizing the SA/RPD body of theory, and resulted in a computational architecture called COGNET [35]. While this architecture proved successful in modeling human performance in work-tasks, it became clear that the model and behavior were unable to represent or reproduce the higher-level complexities of human social behavior and social intelligence. More recently, the research team focused on developing a cognitive architecture called PAC, based on narrative reasoning and cognitive theories of personality [24,25,33]. While PAC proved able to model and predict complex interpersonal behavior in off-line simulations, the translation of this to real-time situations proved daunting. Specifically, it became clear that to carry out narrative reasoning in real-time, the narrative reasoning knowledge elements required access to a dynamic, and more detailed, representation of the changing understanding of the problem context at lower levels of abstraction. This required, in the end, adding much of the SA/RPD mechanisms for building context from COGNET into the narrative-based mechanisms in PAC. The addition of these mechanisms fell far short of true integration, however, in that a common theoretical framework for such an integration was lacking. The framework described below was developed to meet this need.

A. Framework for Integration

The main idea underlying this integration is that what SA/RPD and narrative reasoning theories implicitly or explicitly refer to as the understanding or awareness of *context* is really a momentary "snapshot" of fundamental processes integrating multiple sources of information about the natural and human (i.e., social) aspects of the environment. This process of *context development* is *constructive*, *self-organizing*, operates at *multiple discrete levels of abstraction* which generally involves *increasing time-scales* across levels. These four key features are defined as follows:

- *Constructive* -- consists of constituent elements that, through their interaction, build a symbolic representation, the momentary content of which we may consciously recognize as the current context.
- S*elf-organizing* -- the constituent elements operate independently but follow principles or rules of operations that are organic to the human information processing design, such that a consistent and self-regulating process (of context development) emerges.
- *Operates at multiple-discrete levels of abstraction* -- the symbolic representation which is built and maintained has distinct layers of structure which reflect levels of understanding that each incorporate a broader scope of information about the environment but in correspondingly increasingly abstract terms that include salient and diagnostic attributes, with links to lower levels of abstraction where more detailed (but less integrated) information is maintained. These levels equally organize the constituent processing elements that build the context representation as much as they organize the representation itself. In this initial formulation of the framework, there are four levels corresponding to the three hierarchical levels of Situation Awareness (Perception, Comprehension, Projection) and one higher level of Narrative Understanding which integrates the other three. We thus call the framework the NIM (Narratively-Integrated Multilevel framework.

- *Involves increasing time-scales across levels* -- each increasing level of abstraction deals with a broader scope of events (from perceptual events at the lowest level to narrative units at the highest level). As that scope increases, the general time-scale of events similarly increases. For example, perceptual events, such as those tracking locations of a (single) moving object, are very dense in time and result in repeated updates to perceptual level information in the context representation. At higher levels, updates typically occur less frequently, as many lower level changes are needed to create a significant or meaningful update. Narrative pacing, the highest level, typically is the slowest, as a great deal of action in the environment is typically aggregated into a single narrative unit. This relationship of increasing time scale and increasing scope is very similar to the concepts presented in Newell's timescale of human action [19: Figure 3-3]. Thus, the amount of processing would tend to be much greater at lower levels, though the scope and usefulness of the information in the representation would tend to be much broader at higher levels. However, because of the constructiveness feature, the highest level cannot be constructed without all the processing involved in building and maintaining the lower levels.

The dynamics of the process are moved forward both by sensory information (on the external world), physical actions (taken in the external world), and internal sources of information that can be termed knowledge elements. In the NIM framework, the context representation is constantly being manipulated in different ways by knowledge elements (KEs) that themselves are activated by *externalities* (in the form of sensations and/or physical actions), or by *internalities* (in the form of patterns of information within the declarative representation or associations to past experiences). Thus, the various knowledge elements construct and maintain the context representation in a self-organizing way, without any explicit starting or stopping (or other control) mechanism.

B. Computational View of the NIM Framework

As a process, context development is an example of, and can be computationally modeled using, Selfridge's Pandemonium architecture [29], which has been highly influential in many branches of cognitive science and artificial intelligence over the last half century. In a Pandemonium-style model of the context process, a hierarchical declarative representation of context is the central feature, and elements (chunks) of knowledge are spontaneously activated (and compete for attention) by patterns of information and dynamic changes to this declarative representation. Each element of knowledge changes the declarative context representation (making it a *representation-building knowledge element*), either by creating new information, or by adding, replacing or deleting information, At any point in time, the DMs understanding of the context is the current content of the declarative context knowledge structure. The context development process is pictured in Figure 1.

It can be argued that a background process that develops and maintains an understanding of context is a highly adaptive characteristic of human beings, because it provides the individual a constantly available basis for interacting with the environment. The representation-building knowledge elements that construct the context representation reflect both individually acquired expertise and culturally-transmitted understanding of the local or domain-specific environment, so the context representation is not only always available, but also encodes information that experience (individual and collective) has shown to be useful in those environmental interactions. Ultimately, it is through its ability to support effective actions and interactions in the natural and social environment that the value of the context process is realized.

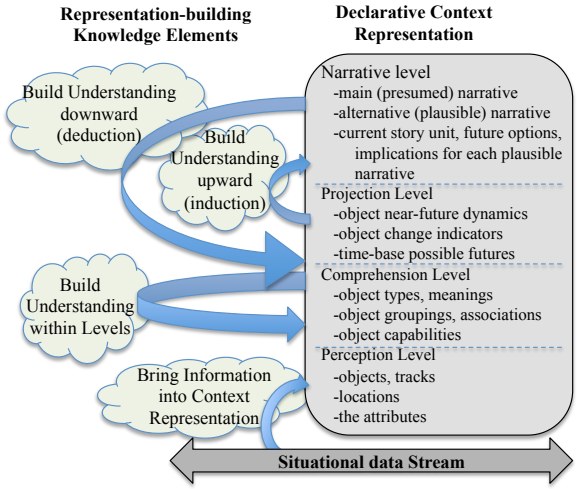


Figure 1. Context Development Process

III. Context and Decision-Making

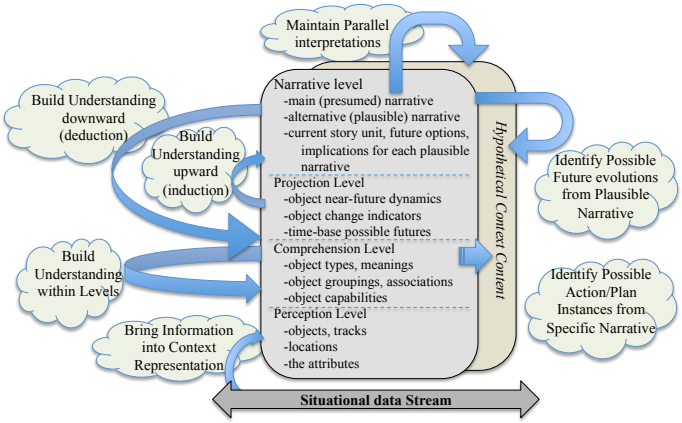
Research into decision-making has explored some of the ways in which the context representation supports decision-making. The RPD model, most specifically, has demonstrated that expert DMs are in many cases able to select an action or adapt a pre-existing action plan to a specific situation based on the patterns of information in the context model. The patterns of information prime a specific decision (course of action) without requiring intervening deliberative processes. More analytical decision processes, in contrast, involved multiple reasoning steps that manipulate the context representation to construct, rather than derive, a plan or specific action. Across this full continuum of analytical to automatized decision making, (often called the Cognitive Continuum, see [38]) the same process is occurring. Knowledge elements derive or construct decision options and courses of action by manipulating and operating on the information in the context representation. These can be called decision-development KEs.

In light of the above discussion on context development, the decision-development KEs can be seen as are analogous, to representation building KEs. Both use the information in the context representation, but the representation-building KEs use it to create changes to the context representation, while the decision development KEs instead use it to reason toward actions to be taken in the external environment.

To some degree, the preceding begs the question "what is

decision-making?" For purposes here, decision is used broadly to refer to the processes by which purposive actions are selected or constructed, whether or not there is a conscious awareness at the time that a decision is being made. This is broadly in line with RPD theory which notes that the RPD process typically renders what appears, to a novice or outsider, to be a difficult decision, as simply an obvious or automatic action to the expert.

One additional feature needs to be added to the NIM framework to describe or model the relationship of the context-development process to the decision-making process. That is the notion of hypothesizing – constructing and manipulating alternative descriptions or relationship sets for part or all of a context representation, typically by creating hypothesized representations of future contexts that might result from contemplated decisions or actions. For context to support decision making, there needs to be proxy representations of context, in which decision-development KEs can use to construct and assess potential decisions and actions. This space, unlike the context representation, is not an internal model of the external situation, but is rather a hypothesized representation of it as it might be, if potential decisions and actions were taken. This allows such decision-development KEs to maintain alternative multi-level representations of an evolving situation, or project forward possible decisions or actions based on a narrative interpretation or course of action being considered. Figure 2 expands Figure 1 to show how decision-development KEs and hypothetical context representations extend the context development process to support dynamic decision-making of all kinds.



Fiure 2. Context Processes Supporting Decision Processes

IV. Context and Decision Support

The cognitive process of context development and maintenance is common to all human adults, just as is the process by which context understanding is used to make decisions and construct actions in the external environment. The environments in which these human capacities evolved were relatively bounded and unfolded in time scales generally in line with human information processing. However, this began to change in historical times, as social and technological complexity rapidly increased. Since the start of the electronics and computer age, human DMs find themselves increasingly embedded in complex environments in which the speed and complexity of events greatly outstrip human cognitive abilities. Real-time decision-making domains such as military command and control or management of large-scale industrial processes bring environments in which it is essentially impossible for an unaided DM to fully understand the context in which actions must be taken.

The preceding half century has seen increasingly sophisticated efforts to support and augment human decision-making. Research to understand human cognition has been stimulated by the need for more effective decision-support, and has driven the evolution of decision support. In particular, it has resulted in an approach (termed cognitive engineering) to designing decision support systems, based on designing the systems to integrate well with the ways in which humans perceive, think, and act.

The NIM context-development view offers a new basis for cognitive engineering of decision support systems. The framework shows how multiple levels of context understanding are simultaneously developed and maintained, and are also simultaneously used to identify opportunities for action and for action options. This suggests a way to *design* decision support, in which the support system develops its own context representation (based on a model of the human context-development process), and applies this model to develop decision/action information at multiple levels of abstraction. Further, such a system can both provide its context representation to the DM as representational support, and provide its decision/action information to the DM as decision support. Because it is expressed in fundamentally computational terms, the NIM framework suggests a way to *develop* the context and decision models that such a support system would require.

Before providing a brief example of how this might work, we note two other interesting characteristics of the NIM framework with regard to the application areas of interest to this conference. The first is in the area of *human-machine integration*. Substantial research and engineering effort has been devoted to automating the process by which a human operates a continuous-control system, such as a vehicle or power plant. In between manual control and full automation, however, are many approaches to partial automation that structure the engineering space. All generally fall under the concept of supervisory control (originated by Sheridan and Johannsen, [30]). In supervisory control, many or all the functions of manual control are automated within a space of options or assumptions. The human may turn over control to those automated functions to free time and attention for other activities, but only while supervising the automation for changes in the underlying options or controls. When such changes occur, the operator will need to either resume manual control and/or modify the settings on the automation. The autopilot on a manned aircraft is an example of this process. Supervisory control is a human-machine integration concept, because it frames how the interconnection between human and automated system components is engineered. If a system allows only supervisory control, then it can be labeled as having pure supervisory control. If, however, the human can assume direct control as well as supervisory control then the system can be said to have mixed mode control. NIM context

development allows control processes to be framed and embedded within it. This can be done by considering control to be a continuous analog of (discrete) decision-making, and mapping the forms of control to the level of abstraction on which they rely in the context representation. Manual control, for example, involves context understanding largely at the perceptual level and significance levels. Supervisory control, in contrast, involves context understanding at the significance and projection level. Control at the highest levels of abstraction are not widely discussed in the human-machine integration literature, but they could be described as situational control or narrative control, in which control is only applied to choice of narrative interpretation and choice of narrative units, with all lower level control being automated. This relationship is pictured in Figure 3, discussed below.

The second is an interesting correspondence between the context development NIM view of context development and *military models of decision making*, particularly the military decision making model known as the Observe-Orient-Decide-Act or OODA Loop, first created by Boyd in the 1980s [2,23]. It teaches military DMs to view decision-making as an on-going process, in which situational understanding, achieved by careful observation (Observe) and interpretation (Orient), lead to courses of action (Decide) that are implemented and have effects on the situation (Act). These effects then change the situation (as do actions of the opponent and other non-combat processes), requiring a new or ongoing process of observation and interpretation. In addition to it being widely used in military education and doctrine development, the four components of the loop map very closely to the ways in which context information is used in the NIM framework. That is, the activities of the:

- representation-building KEs that effectively import sensed information into the context representation corresponds to the Observe stage;
- representation-building KEs that integrate context information and build context understanding through and across levels corresponds to the Orient stage;
- decision-development KEs that identify potential courses of action corresponds to the Decide stage; and
- decision-development KEs that construct the details of action plans and physically implement those plans maps to the Act stage.

V. A CONCEPTUAL EXAMPLE

A notional example is provided below to demonstrate the potential application of the NIM framework. The example focuses on the management and control of multiple uninhabited vehicles (UxVs). Such groups of vehicles can be used in diverse missions ranging from post-disaster search and rescue, to battlefield intelligence collection and tactical interdiction. The framework was used to map out the context process in this domain, and to link it to support for both the Observe/Orient stages of the OODA loop and the Decide/Act stages. The result is pictured in Figure 3.

The figure is organized top-to-bottom into four stacked bands that represent the four levels of context representation. The figure has a left-to-right structure as well. In the center of the figure is a box that represents the dynamic context development process, as it would be performed by a computational model. That box is divided into two columns, with the left depicting the various levels of context representation, and the right representing the corresponding representations constructed to develop decision and action plans from the context representation. These two columns correspond to the Observe/Orient and Decide/Act phases of the OODA loop.

On the immediate left of the context development box is a column that represents the representation-building KEs. These KEs both dynamically build/maintain the context representation, and push information to the next (on the left) column as support for the human DM's understanding of the context. On the immediate right of the context-development box is a column that represents decision-development KEs that dynamically build/maintain representation of decisions and actions based on current context dynamics, and that push information to the next (on the right) column as support for the human DM's selection and instantiation of action options. Thus, the entire left side of the figure represents support for the OO parts of the OODA loop, while the entire right side represents the support for the DA parts.

Below the lowest level of context is a black bar that represents the environmental interfaces decision system (human augmented by context-driven support). In the case of multi-UxV command and control, these environmental interfaces would be with various sensors and information streams from the UxVs being controlled.

In Figure 3, the context-development process builds upward from perceiving basic situational information (Level 1) through identifying the significance of the elements (Level 2) and projecting the capabilities of key elements forward into the future (Level 3). From that, the lower level information is fit into stories and understood in the context of the narrative of the current mission (Level 4). The right-most column of Figure 3 then depicts the reasoning activities that the context-based decision support model is performing to take action in the environment and accomplish the mission. At the highest level, the model may revise or refine the current story narrative, and update it in terms of his/her evolving lower level context understanding. As the action proceeds to the point that a choice must be made between possible 'next' narrative units, the model makes use of the current context to choose a possible path forward (through the current narrative space), and conveys it to the human DM. If the DM concurs, the model could translate that general narrative step into specific local action plans (e.g., creating new waypoints, altitude, sensor-settings, etc.).

Additional detail can be seen by more closely examining the two columns labeled "Observe/Orient" and "Representation Building KEs" from bottom to top. Figure 3 shows that the:

- Object representations of information from sensors and/or data streams are created as the lowest levels of context information, using sensory KEs (e.g., monitoring sensor feeds looking for new data, which are then processed to create a new track object or update an existing one).

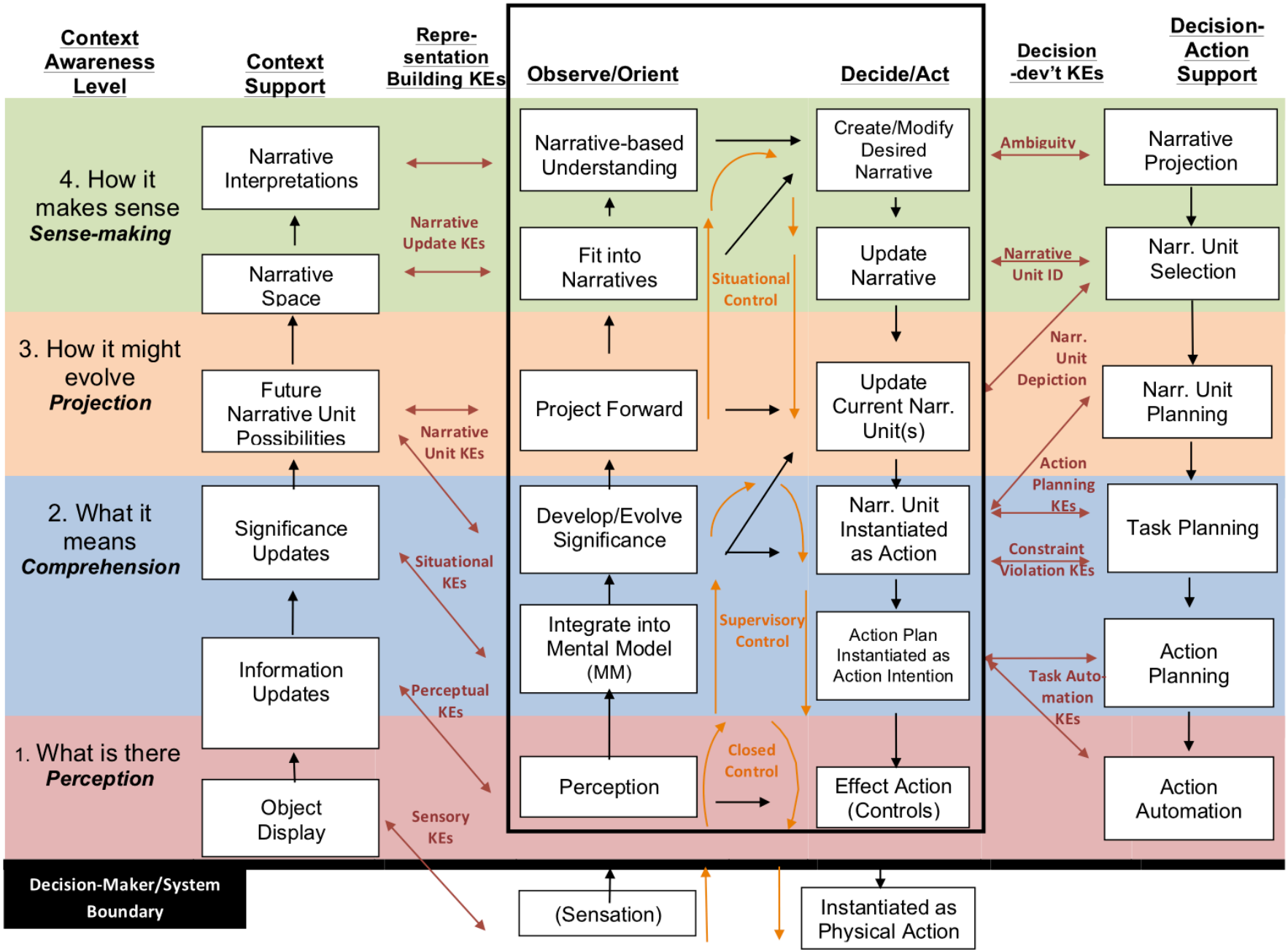


Figure 3. Conceptual Example of Framework Applied to Multiple UxV Control

- Declarative context representation is built and updated from the primitive object representations by perceptual KEs that construct a multi-level structure with built-in semantic significance regarding the levels; information is created and modified as elements of meaning are inferred or created for them. Initially, the perceptual KEs look for information with specific kinds of meaning (e.g. waypoints, vehicle locations) and place them in the context structure.
- Context representation updates happen continuously as situational KEs combine information from multiple places in the context representation. For example the appearance of a hostile radar emission (created by a perceptual KE) might trigger a situational KE to examine all UxV tracks and infer which one(s) may have been detected, and to add a 'likely detection by hostile' annotation to that UAVs information on the context representation. That changemay, in turn, trigger another KE to add a 'need to evaluate' annotation on the track to stimulate examination of its altitude or flight path.
- Narrative updates happen as changes in the dynamic content trigger Dynamic Narrative KEs to offer evidence on whether the narrative may have changed from one narrative state to another. For example, the preceding hostile radar may, if expected, activate a Dynamic KE to post evidence that the narrative may have moved from an 'ingress' phase to an 'in hostile airspace' narrative state.
- Finally, Narrative Space Update and Narrative Interpretation are made as Narrative Update KEs weigh evidence for and against a transition across narrative units. If posted evidence outweighs posted counter-evidence above a threshold, then a Narrative Update KE may be triggered to update the story narrative to reflect that narrative-state transition. Other Narrative Update KEs can be triggered by very anomalous information that may activate narrative re-examination. For example, if the story narrative were about a reconnaissance in a demilitarized area, the presence of the sudden hostile radar detection may trigger a Narrative Update KE that would look for other narratives that might incorporate this fact which does not 'make sense' in the baseline narrative. That KE might suggest re-examination of the data against stories of outbreak of hostilities or new insurgent activity as alternative stories.

This example is intended to point out how the NIM view of context development as an ongoing and core cognitive process can act as an integrating element for advanced decision support systems. Moreover, the example suggests how the framework can be further applied to integrate the design of human-systems integration and to translate the cognitive and technological issues into widely accepted military concepts such as OODA that can support the transition of such advanced decision support systems into operational use.

VI. DISCUSSION

This NIM framework presented here is built on the premise that human decision makers approach and resolve a decision based on their understanding of the situational context of that decision. When the decision maker is operating within a class of situations whose structure he/she understands very well, her or his internal context model will be rich and organized at multiple interconnected levels of abstraction. Such a NIM context representation provides insights at each level of abstraction – from low-level immediate details to long-term high-level story-structures – and enables mechanisms that allow situational interpretations and decision options to be considered at each level in an integrated way. A key implication of this research is that any externally provided (i.e., computational) decision support information will be evaluated and considered by the decision maker in terms of his/her own internal context understanding. Thus, from a cognitive engineering perspective, any and all decision support components, algorithms, etc., should present their results in terms of the decisions makers' context model, and should ideally be designed to be presented in such terms from the start. As implied here, one way in which this can be done is for the computational decision support system to build and maintain its own context representation, strongly modeled to mimic the context representations created and maintained by expert decision-makers in the domain.

In conclusion, we offer thoughts on the validation of the NIM framework, and the ways in which semantic technologies can be used to implement the NIM framework.

Validation. The difficulties of validating models of cognitive processes, which are inherently unobservable, are well discussed in the literature. Validation in cognitive science is, in philosophy of science terms, typically limited to standards of sufficiency (i.e., can a model explain all the data) rather than necessity (i.e., only that model do so). Prolonged validation studies for very fundamental constructs (such as short term and working memory, see [3]) have be approached with experimental studies, but, even there, competing models remain even after decades of experimentation. For higher level models of cognitive processes that are not biological but rather which emerge from embodied experience in the world (such as the NIM framework for context), the validation problem is that much more difficult. Ultimately, we believe that validity can be locally approached with specific domains and specific populations of decision makers, using established cognitive science data collection methods such as thinking aloud data collection, situationally-adapted verbal probes, and retrospective interviews. Through such domain-based explorations, incremental local validation may be achieved, which may lead to broader acceptance over time.

Semantic Technologies and NIM Implementation. Semantic technologies (the topic of this conference) can form the core of a computational system that implements a domain-specific model using the NIM framework. In fact, initial efforts to date have made increasing use of these, particularly the Resource Description Framework (RDF) semantic representation. While the earlier COGNET software used a custom-coded blackboard representation to create the lower three levels of the NIM declarative context representation, the most recent versions of the PAC software have moved toward a implementing the declarative context representation fully in RDF. Current research to integrate these two computational models is also focusing on RDF for all levels of context

representation. The semantic RDF representation is then manipulated by KEs implemented as production mechanisms, sometime gathered into more complex require structures that chunk multiple reasoning elements into a unitary NIM KEs.

REFERENCES

[1] Bex, F., Van den Braak, S., Van Oostendorp, H., Prakken, H., Verheij, B., & Vreeswijk, G. (2007). Sense-making software for crime investigation: how to combine stories and arguments?. *Law, Probability and Risk*, *6*(1-4), 145-168.

[2] Boyd, J. R. (1987). Organic design for command and control. *A Discourse on Winning and Losing*.

[3] Card, S., Moran, T., and Newell, A. (1983) *The Psychology of Human-Computer Interaction*. Mahwah, NJ: Erlbaum.

[4] Dyer, M. G. (1983). *In-depth understanding: A computer model of integrated processing for narrative comprehension*. MIT press.

[5] Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *37*(1), 32-64.

[6] Endsley, M. R. (1997). The role of situation awareness in naturalistic decision making. *Naturalistic decision making(A 97-39376 10-53), Mahwah, NJ, Lawrence Erlbaum Associates, Inc., , 1997,*, 269-283.

[7] Endsley, M. R. (2000). Flight crews and modern aircraft: In search of SA.*Royal Aeronautical Society*, *12*.

[8] Endsley, M. R., & Connors, E. S. (2008, July). Situation awareness: State of the art. In *Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE* (pp. 1-4). IEEE.

[9] Feng, Y. H., Teng, T. H., & Tan, A. H. (2009). Modelling situation awareness for Context-aware Decision Support. *Expert Systems with Applications*, *36*(1), 455-463.

[10] Graesser, A. C., Singer, M., & Trabasso, T. (1994). Constructing inferences during narrative text comprehension. *Psych. Rev*, *101*(3), 371.

[11] Hair, D. C., & Pickslay, K. (1993). *Explanation-based reasoning in decision support systems.* Technical Report. NAVAL COMMAND CONTROL AND OCEAN SURVEILLANCE CENTER RDT & E DIV. SAN DIEGO CA.

[12] Hutchins, E. (1995). *Cognition in the Wild* (Vol. 262082314). Cambridge, MA: MIT press.

[13] Klein, G. (2008). Naturalistic decision making. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *50*(3), 456-460.

[14] Klein, G. A. (1993). A recognition-primed decision (RPD) model of rapid decision making. *Decision making in action: Models and methods*, *5*(4), 138-147.

[15] Klein, G. A., Calderwood, R., & Clinton-Cirocco, A. (1986, September). Rapid decision making on the fire ground. In *Proceedings of the Human Factors and Ergonomics Society annual meeting* (Vol. 30, No. 6, pp. 576-580). SAGE Publications.

[16] Klein, G., A., Calderwood, R., & MacGregor, D. (1989). Critical decision method for eliciting knowledge. *IEEE Transactions on Systems, Man, and Cybernetics, 19*, 462-472.

[17] Miller, L. C., & Read, S. J. (1991). On the coherence of mental models of persons and relationships: A knowledge structure approach. In G. J. O. Fletcher & F. Fincham (Eds.), Cognition in Close Relationships. (pp. 69-99). Hillsdale, NJ: Erlbaum.

[18] Morrison, J. G., Kelly, R. T., & Hutchins, S. G. (1996, October). Impact of naturalistic decision support on tactical situation awareness. In *Proceedings of the Human Factors and Ergonomics Society annual meeting* (Vol. 40, No. 4, pp. 199-203). SAGE Publications.

[19] Newell, A. (1990) Unified Theories of Cognition. Cambridge: Harvard Univ. Press

[20] Niu, L., & Zhang, G. (2008, December). A model of cognition-driven decision process for business intelligence. In *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on*(Vol. 1, pp. 876-879). IEEE.

[21] Pennington, N., & Hastie, R. (1993). The story model for juror decision-making. In R. Hastie, (Ed). *Inside the juror: The psychology of juror decision-making. Cambridge series on judgment and decision-making*. (pp. 192-221). New York, NY, US: Cambridge University Press,

[22] Pennington, N., & Hastie, R. (1993). *The story model for juror decision making*(pp. 192-221). Cambridge University Press.

[23] Polk, R. B. (2000). A Critique of the Boyd Theory-Is it Relevant to the Army?.*Defense Analysis*, *16*(3), 257-276.

[24] Read, S. J., Miller, L. C., Kostygina, A., Chopra, G., Christensen, J. L., Corsbie-Massay, C., Zachary, W., LeMentec, J. C., Iordanov, V., & Rosoff, A. (2007). The Personality-enabled architecture for Cognition. In A. Paiva & R. Picard (Eds.). Affective Computing and Intelligent Interaction 2007. Springer-Verlag.

[25] Read, S.J., Miller, L. C., Monroe, B., Brownstein, A., Zachary, W., LeMentec, J. C., & Iordanov, V. (2006) A Neurobiologically Inspired Model of Personality. In J. Gratch et al. (Eds.), I*ntelligent Virtual Agents 2006 (IVA 2006)* (pp. 316 – 328), Lecture Notes in Artificial Intelligence, 4133. Berlin: Springer-Verlag.

[26] Riessman, C. K. (1993). *Narrative analysis* (Vol. 30). SAGE Publications, Incorporated.

[27] Schank, R. C., & Abelson, R. P. (1995). Knowledge and memory: The real story. In R. S. Wyer, Jr. (Ed.), Knowledge and memory: The real story. Advances in social cognition, vol. 8 (pp. 1-85). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

[28] Schank, R. C., Abelson, R. P. (1977). Scripts, plans, goals, and understanding: An inquiry into human knowledge structures. Hillsdale, NJ: Erlbaum.

[29] Selfridge, O. G. (1958). Pandemonium: a paradigm for learning in mechanisation of thought processes.

[30] Sheridan, T. B., & Johannsen, G. (Eds.). (1976). *Monitoring behavior and supervisory control*. New York: Plenum Press.

[31] VanLehn, K. (1996). Cognitive skill acquisition. *Annual review of psychology*, *47*(1), 513-539.

[32] Zachary, W., Le Mentec, J.-C.. Miller, L.C., Read, S. J., & Thomas-Meyers, G. (2005a). Human behavioral representations with realistic personality and cultural characteristics. *Proceedings of the Tenth International Command and Control Research and Technology Symposium*, McLean, VA.

[33] Zachary, W., LeMentec, J-C., Miller, L. C., Read, S. J., & Thomas-Meyers, G. (2005b). Steps toward a Personality-based Architecture for Cognition. *Proceedings of the 2005 Conference on Behavioral Representation in Modeling and Simulation*, Los Angeles, CA.

[34] Zachary, W. & Ryder, J. (1997) Decision Support: Integrating Training and Aiding. In M. Helander, T. Landauer, and P. Prasad, Eds., *Handbook of Human Computer Interaction, 2nd Edition.* Amsterdam: North Holland, pp. 1235-1258.

[35] Zachary, W., Ryder, J., Stokes, J., Le Mentec, J.-C., Santarelli, T. (2005) A COGNET/iGEN Cognitive Model that Mimics Human Performance and Learning in a Simulated Work Environment. In K. Gluck & R. Pew (Eds.) *Modeling Human Behavior with Integrated Cognitive Architectures.* Mahwah, NJ: LEA.

[36] Gentner, D., & Stevens, A. L. (Eds.). (1983). *Mental models*. Psychology Press.

[37] Hampton, G. (2009). Narrative policy analysis and the integration of public involvement in decision making. *Policy sciences*, *42*(3), 227-242

[38] Hammond, K. R, Hamm, R. M., Grassia, J., & Pearson, T. (1987). Direct compar- ison of the efficacy of intuitive and ana- lytical cognition in expert judgment. *IEEE Transactions on Systems, Man, and Cybernetics, 17(5),* 753–770.

Towards a Context-Aware Proactive Decision Support Framework

Benjamin B. Newsom, Jr[1]
Next Century Corporation
Columbia, Maryland, USA

Ranjeev Mittu[2]
Naval Research Laboratory
Washington, DC, USA

Ciara Sibley[2]
Naval Research Laboratory
Washington, DC, USA

Myriam Abramson[2]
Naval Research Laboratory
Washington, DC, USA

[1]ben.newsom@nextcentury.com, [2]{firstname.lastname@nrl.navy.mil}

***Abstract*—The problem of automatically recognizing a user's operational context, the implications of its shifting properties, and reacting in a dynamic manner is at the core of mission intelligence and decision making. Environments such as the OZONE Widget Framework[1] provide the foundation for capturing the objectives, actions and activities of the mission analyst and decision maker. By utilizing a "context container" that envelops an OZONE Application, we hypothesize that *action* and *intent* can be used to characterize user context with respect to operational modality (strategic, tactical, opportunistic, or random). As the analyst moves from one operational modality to another, we propose that information visualization techniques should adapt and present data and analysis pertinent to the new modality and to the trend of the shift. As a system captures the analyst's actions and decisions in response to the new visualizations, the context container has an opportunity to assess the analyst's perception of the information value, risk, uncertainty, prioritization, projection and insight with respect to the current context stage. This paper will describe a conceptual architecture for an adaptive work environment for inferring user behavior and interaction within the OZONE framework, in order to provide the decision-maker with context relevant information.**

Keywords—context-driven; decision-making; dynamic modeling; operational modality; temporal reasoning

I. INTRODUCTION

Today's warfighters operate in a highly dynamic world with a high degree of uncertainty, compounded by competing demands. Timely and effective decision making in this environment is increasingly challenging. The phrase "*too much data – not enough information*" is a common complaint in most Naval operational domains. Finding and integrating decision-relevant information (vice simply data) is difficult. Mission and task context is often absent (at least in computable and accessible forms), or sparsely/poorly represented in most information systems. This limitation requires decision makers to mentally reconstruct or infer contextually relevant information through laborious and error-prone internal processes as they attempt to comprehend and act on data. Furthermore, decision makers may need to multi-task among competing and often conflicting mission objectives, further complicating the management of information and decision making. Clearly, there is a need for advanced mechanisms for the timely extraction and presentation of data that has value and relevance to decisions for a given *context*.

To put the issue of context in perspective, consider the fact that nearly all national defense missions involve Decision Support Systems (DSS)—systems that aim to decrease the cycle time from the gathering of data to some operational decision. The proliferation of sensors and large data sets are overwhelming DSS's, as they lack the tools to efficiently process, store, analyze, and retrieve vast amounts of data. Additionally, these systems are relatively immature in helping users recognize and understand important context (i.e. cues). The next generation systems must leverage predictive models to enable Proactive Decision Support (PDS). These systems will need to understand and adapt to user context (missions, goals, tasks). By aligning the data with the user in the appropriate context, we hypothesize that more relevant information can be provided to the user i.e., likely to be of higher value for decision making. The key challenges, therefore, are to not only model the user's decision-making context, but to recognize when such context has shifted. With regard to Figure 1, we hypothesize that concepts associated with PDS closely align with Prescriptive Analytics (i.e., understanding and modeling decision trajectories and the relevant information necessary for those decisions).

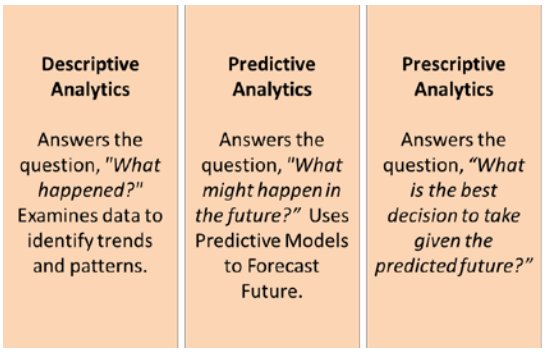


Figure 1: Comparison of different forms of Analytics

[1] http://www.owfgoss.org

The problem of automatically recognizing / inferring user context, understanding the implications of its shifting properties, and reacting in a dynamic manner is at the core of mission intelligence and decision making. An environment such as the OZONE Widget Framework provides the foundation for capturing the objectives, actions and activities of the mission analyst/decision maker. By utilizing a "context container" that envelops an OZONE Application, we can capture both action and intent which allows us to characterize this context with respect to its operational modality (strategic, tactical, opportunistic, or random) – Figure 2 (*Visual Analytics* representation).

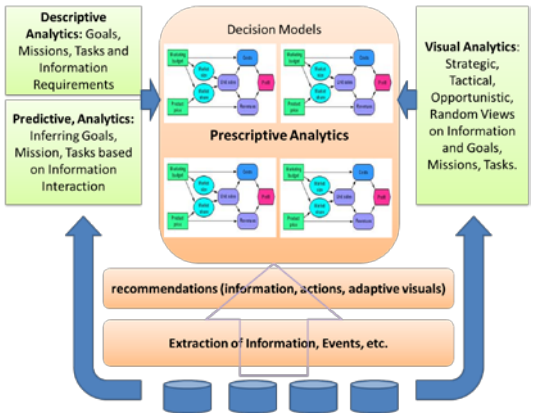


Figure 2: Context understanding in relation to Analytics

Context is fluid over time, and the relative mix of strategic vs. tactical vs. opportunistic actions or activities is also changing. Knowing the time frame and distribution of activities gives us insight into the analyst's changing operational modality. A temporal storage approach, such as a Context-Aware Memory Structure (CAMS), provides the basis for comparison of the "current" decision stage against prior stages and is used to predict phase shift.

Methods for understanding user context can be found in logic-based or probabilistic Artificial Intelligence (AI) approaches under Predictive Analytic Methods, or through more traditional methods based on Descriptive Analytics. Using a Descriptive Analytics approach, models can conceivably be developed that map missions, goals and tasks to information requirements in order to represent "decision context". With regard to deriving context within the Predictive and Visual analytics models, the challenging questions become: Can a user's decision context be modeled, based upon, information seeking, interaction, or analysis patterns [1]? What research can be leveraged from the AI community (plan recognition) to infer which decision context (model) is active? Can we reason about which decision context (model) should be active? What similarity metrics enable the selection of the appropriate model for a given context? Can we recognize context shift based on work that has been done in the Machine Learning community with "concept drift", and how well does this approach adapt to noisy data? The emphasis for the paper will be on the ***Visual Analytics*** representation for understanding context, but the questions span across ***the Predictive Analytics*** representation as well.

In Section II, we provide a notional operational example to guide the framework discussion. In Section III, we describe the APTO system architecture. In Section IV, we briefly describe the idea of Context Container for the APTO framework. In Section V and VI we describe the Context Aware Memory Manager and context shift recognition. In Sections VII, VIII, IX, and X we discuss the adaptive visualization informed through the APTO architecture, event, activity and workflow manager, respectively.

II. Notional Operational Example

Consider the scenario of the intelligence analyst on a *24x7* watch floor (Figure 3). As the analyst moves from one operational modality to another, the information visualization techniques should adapt and present data and analysis pertinent to the new modality and to the trend of the shift. If we can capture the analyst's actions and decisions in response to the new visualizations, the context container may be able to infer the analyst's perception of the information value, risk, uncertainty, prioritization, projection and insight. This information, in combination with the ability to infer the user's current context stage would provide the ability for DSS's to pre-stage information that is tailored to the user's current needs and preferences along a decision trajectory.

Each watch floor is configured and organized to address their unique and specialty mission and intelligence requirements. As such, any solution proposed must be able to adapt and conform to the specific needs of the watch. In Figure 3, we show an example set of watch floor responsibilities with the proposed solution focusing on Analyst activities (3), Cell activities (4), and Watch Officer activities (6).

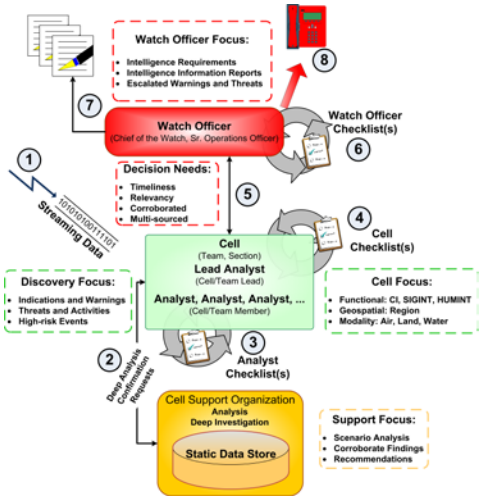


Figure 3: Example Watch Floor Scenario

In general, a watch floor is organized around Cells of responsibility. A Cell (also known as a *Team* or *Section*)

may have only one Analyst with a singular focus, or it may be multiple Analysts with a Lead Analyst (also known as the Cell or Team Lead). A Cell is monitoring and accumulating streaming data (1) to discover indications and warnings about threats and high-risk events in their scope of consideration. Timeliness of analysis and interpretation is critical. The Cell may have a support organization that can perform deep analysis (2) and confirm an Analyst's or Cell's findings. For often-detected indications, the Analyst will have a set of standard operating procedures or checklist (3) of activities they need to perform to reach the decision to escalate the detected event to the next level. In a multi-person Cell, the next level may be the Cell Officer who has their own set of standard operating procedures or checklist (4) of activities that need to be performed to escalate out of the Cell (5).

An event (threat or warning) escalated out of the Cell (5) goes to the Watch Officer who is accumulating information and comparing escalated events to their Intelligence Requirements. Like the Analyst and the Cell, the Watch Officer has a set of standard operating procedures or checklist (6) of activities to perform in response to the combination of escalated events that they are receiving from all of the Cells on the watch floor. The Watch Officer makes the trade-off decisions to only track and log (7) the events (threats) or escalate identified, confirmed, credible threats (8) to the next level.

The watch floor situation has intense analytical problems requiring timely analyses and/or responses. Analytical problems are often sensitive and associated with high stakes for success or failure. In many analytical sub-domains, the objectives of the analysis can be open and shifting, and analysts must sometimes determine for themselves the goals and priorities of their data collection or research. The proposed framework identifies the context in which the events and activities are occurring and provides situational awareness and accuracy up and through the chain of decision makers.

The proposed system architecture should extend and enhance existing mission solutions to include PDS focusing on context shift recognition and staging of the information (or combinations of information) the analyst requires in making the "next" decision. Along with determining the information to be staged, the adaptive work environment needs to react to the context shift and determine the appropriate stage-related information visualization techniques.

To accomplish the objective of inferring a user's context and recognizing context shifts, there are three broad areas of required innovation:

- Capturing context actions and events through normal analyst interaction with OZONE Framework applications.
- Characterizing the user's actions and events along their operational modality (i.e., strategic, tactical, random discovery, and opportunistic discovery), their temporal relationship, and situational objectives.
- Recognizing the change or shift in context through the development of Context Shift Models and predictive analysis.

III. APTO SYSTEM

A. Long Term Goal

In order to create a context-aware adaptive work environment, specific elements such as the memory components, the context manager, and the Activity Manager are necessary for recognizing context and context shift. APTO (Latin for adapt) is a conceptual architecture, shown in Figure 4, that depicts a context-aware environment within the OZONE Widget Framework.

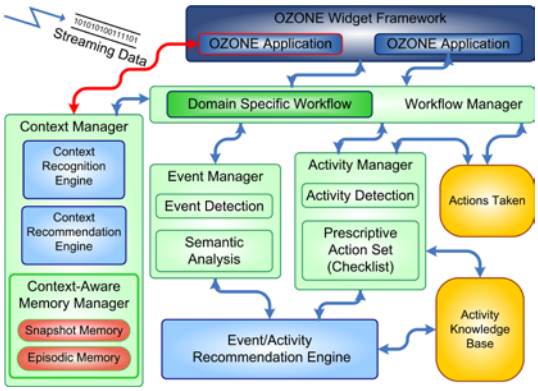


Figure 4: Conceptual APTO Architecture

B. Technical Approach

The premise of our approach is that the combination of an intelligence analyst's OZONE Application (sometimes referred to as widgets) usage pattern and the information being visualized (and how it is visualized) can be used as an indicator of the analyst's context mode. The analyst is viewing all of the situation characteristics through a particular lens searching for strategic insights, tactical clues, opportunistic indicators or the random-scramble searching for the information nugget that connects decision streams together. Through adaptation and innovative extensions to the OZONE Widget Framework, it will be possible to capture traces of user interactions with the widgets, as well as interactions between widgets. We believe this situational capture of the decision making process will form distinctive, predictable patterns of behavior corresponding to the analyst's intent, information value, and prioritization.

IV. CONTEXT CONTAINER FOR OZONE APPS

The concept of a "context container" for OZONE Apps does not exist in the current OZONE Widget Framework. In the overall architecture, it is part of the

interaction between the user experience or presentation layer and the Context Manager. We believe that we can define and create a container or software envelope that would "wrap" an OZONE application (a collection of one or more widgets and data sources) and automatically capture both what the decision was and an indication of why (operational modality) the decision or choice was made. This collection of activity, interaction and/or decisions represents a context vector that would be stored in the Context-Aware Memory Manager.

V. CONTEXT-AWARE MEMORY MANAGER

To model the analyst's context, a learning context memory model (a Context-Aware Memory Manager – CAMS) [2] could be constructed. This model would capture the OWF widget interactions and process them to construct a context memory reflecting the user's regular activity. The concept of a Context-Aware Memory Manager that interacts with the OZONE Widget Framework does not currently exist.

Context memory is a mechanism for retaining and recalling interesting and relevant past experiences or actions [3]. We believe that an analyst's context consists of a striation or mix of strategic, tactical, opportunistic and random actions. In each layer there are a collection of short-term or "snapshot" memories and long-term or "episodic" memories.

A. Snapshot Memory

The snapshot memory (context working memory) processes and stores context attributes from context input vectors. Attributes are stored in Artificial Recognition Balls [4] (ARBs), which describe a certain region around the context attribute —in the case of OZONE Apps it would be the context container— and enables CAMS to perform data compression by eliminating the need for repetition. For example, a particular type of action can be represented by a single ARB instead of all individual actions that occur within the container; every ARB has a resource level *R* associated with it, being an indicator for how frequently it recognizes context attributes. The algorithm used in CAMS is based on the principles of unsupervised and reinforcement learning. Unsupervised learning allows us to construct a system which can cluster input data without any prior knowledge about the structure of every class. Reinforcement learning requires feedback from a trainer. However, an explicit trainer is not desirable in most context-aware systems, therefore an ARB receives positive feedback (stimulation) when context attributes fall within a certain distance from the center, resulting in an increase in its resource level. Negative feedback is introduced by the notion of 'forgetting', which gradually decays all resource levels. For example, actions a user performs less often have their resource level reduced by a decay factor, but every re-occurrence stimulates it again, which enables these actions to remain in memory.

B. Episodic Memory

To capture a significant part of human activity the connections between consecutive events or actions are essential. The snapshot memory is able to capture every individual action, but not the set of actions that comprise a specific decision. As the user is most likely to login/logout, start up an application, etc., those actions have a higher resource level *R*. Once *R* reaches a predefined level, the oft repeated actions are passed from the Snapshot Memory to the Episodic Memory, which captures all individual attribute values between them. The Context Memory Manager component regulates the division of the memory mechanism into Snapshot and Episodic Memory. This division is essential for keeping the complexity of the search space at a manageable level. Without this division all attributes and connections between them would have to be stored in a directed graph in order to detect and capture meaningful consecutive events — which would result in an *NP* complete search problem. Instead, only the attribute vectors between ARBs with a high resource level need to be stored; after the validation of an episode this is reduced to storing only references to ARBs recognizing the attributes in these vectors. The ARBs with a high resource level *R* passed on from the Snapshot Memory are stored in a cache structure.

Initially, the user will be asked to name and validate a new or preliminary episode, bridging the gap between the data representation within CAMS and the real world meaning. An episode is an ordered 3-tuple containing a start ARB, an end ARB and an ordered list of all context vectors encountered. Ideally, in order for the proposed system to diffuse into every day environments, APTO could learn from the human-assisted validation and move towards automatic recommendations for naming and validation. Only frequently occurring episodes would be presented.

VI. CONTEXT SHIFT MODEL AND SHIFT RECOGNITION

We believe that we can create a network model of the ordered 3-tuple activities that represent each of a context mode's three stages: entering a mode, "*in-the-flow*" of a mode, and exiting a mode based on user interaction patterns. These context mode stage models can be compared to a dynamic modeling of the analyst's real-time activities for detecting shifts and flows of focus. Each mode stage (entering, in-flow, leaving) is a combination or mix of the operational modalities (strategic, tactical, opportunistic, or random) within a particular time frame.

This mix is constantly changing as new information is being presented to the analyst. This combination of actions (e.g., 80% strategic, 12% tactical, 6% opportunistic and 2% random) collected from the analyst's interaction with APTO, will provide the context profile for that analyst at that given time. As

they interact with APTO, their profile trend changes, thus their context and items of interest change.

In particular, the user experience activity of "zooming in" on the temporal aspect of streaming data typically characterizes a tactical desire to narrow the focus for an immediate decision. Typically, this behavior is followed by a "zooming out" to take a more strategic view of the information looking for particular clusters of relevant events or activities. Although this is typical, not all analysts operate in the same manner. Our proposed approach is to accommodate an individualized recognition of pattern and transition indicators [5]. By capturing usage patterns and successful episodes on an individualized basis, the system will be able to adapt its shift recognition to the specific analyst. Over time, the patterns accumulated could become the basis for identification of a best practice approach for often repeated situations.

VII. CONTEXT SHIFT-AWARE STAGING AND VISUALIZATION

Our "context shift" goal is to deliver an individual-focused, context-aware component that can feed its analysis and recognition of transition stages to our context-aware components so that they can anticipate and pre-stage data and recommendations. The analyst's "*view of the world*" should adapt to the individual's operational modality (strategic, tactical, opportunistic, or random). This includes recognizing the data sources, widgets and visualization techniques that are applicable to the particular mode. This identification process will rely heavily upon the context container that encompasses and defines the operational characteristics of the OZONE App.

VIII. EVENT MANAGER

The basis for the Event Manager comes from the Event Representation and Structuring of Text (EVEREST) project, sponsored by the Office of Naval Research. It is an SBIR initiative that has developed text analytic technology that crosses the semantic gap into the area of event recognition and representation. The EVEREST system searches for mappings to a semantic event model, interactively suggesting evidence for the occurrence of whole or partial events for human analysis and reporting. The semantic targeting approach extends the ideas of Open Information Extraction [6], Event Web [7], Semantic Web [8], and the OZONE Widget Framework. EVEREST's event-centric approach is critical for generating narratives that confer meaning upon large, complex, uncertain, and incomplete data sets.

A. Event Detection

The event detection component is based on an Open Information Extraction (Open IE) [9] approach. Open IE systems distill huge volumes of text into a long list of tuples (two entities and one relation that binds them) without asking a human for examples of those relations first. We consider each entity→relationship→entity tuple to be an event assertion. The extractions of assertions from the text are entirely lexical in nature. The assertion extraction utilizes Stanford's core NLP libraries and makes use of a *part-of-speech* tagger (annotator) and noun phrase "chunker." To locate the word in the vicinity of the two nouns (or noun phrases) that mostly likely intended to express their relationship, the detection algorithm employs a technique known as conditional random fields. In essence, this is a statistical model that is sensitive to its lexical context.

B. Prescriptive Event Recognition

The Prescriptive Event Recognition component comprises an event semantic model (metadata and list of assertions) and event inference engine that compares predetermined Target Event models with Reports (detected metadata and list of assertions) in the input stream. The event semantic model is based on Wasterman and Jain [10].

The event inference engine is a mixed-initiative application, i.e., one with a human in the loop, which compares extracted assertions against a prescribed model using a rules-over-graphs approach. The key idea is that many inferencing algorithms used by logic-based AI systems can be heuristically approximated by a much simpler and more efficient system based on graph-matching algorithms. The assertions associated with a Target Event are modeled as a graph of nodes and edges. The nodes are the entities of the tuple. The edges are the relationships between the entities. Similarly, the event assertions detected in the incoming data stream are modeled as a graph of nodes and edges. The graphs are compared for shape, structure, directionality of the edges, content (metadata) of the nodes, and content (metadata) of the edges. Each comparison is scored or ranked to determine how closely the detected event assertion matches the Target Event.

The Prescriptive Event Recognition component offers a list of assertions that are candidate matches for a Target Event. The initial list of candidate assertions are ranked by the inference engine based on its searches for class, instance, and relation isomorphisms between all of the assertions and its semantic event models; an assertion with a closer resemblance will find itself higher on the list. The informational value of the assertion—whether it would fill a central node or an outlier in the graph—will influence the rank as well. The user can decide to accept (or reject) the assertion after consulting his own knowledge, source documents, or other materials. This process could be utilized to fill in missing parts of a graph, which in turn could be utilized by the system to uncover new pieces of information, and this cycle would continue until a target concept has been proven.

IX. ACTIVITY MANAGER

The Activity Manager is focused on activities that are occurring inside the APTO architecture. It interacts with

OZONE Applications via the context container, with the Context Manager module, and the Actions Taken repository.

A. Action Detection

The Action Detection component interacts with OZONE Applications via the context container and the Actions Taken repository. It monitors all of the activities occurring within APTO and identifies actions of interest to the Domain Specific Workflow and routes these actions to the Prescriptive Action component.

B. Prescriptive Action Set

The Prescriptive Action component comprises an action semantic model (metadata and a list of assertions) and an activity inference engine that matches predetermined Action Sets (checklists) with Events (detected metadata and a list of assertions) and Actions Taken. Similar to the common event model proposed by Wasterman and Jain [10], the action semantic model contains temporal elements (the time horizon over which the action should occur), spatial elements (the geographic location where the action should occur), structural elements (the set of action assertions, process steps, or checklist items that need to occur), informational elements (the actor that should perform the action), and causal elements (the set of event assertions that caused this particular action model to be selected).

C. Suadeo Recommendation Engine

Suadeo (Latin for recommendation) is a prototype context-aware, model-driven, recommender system that utilizes "static" persistent data and streaming data as the basis for deriving its recommendations. The intent of the Suadeo prototype is to be a hybrid recommender system that is context-aware with the context model being defined along multiple dimensions such as person, place, time, and incident. The recommendation engine is driven from a graph-based analysis of the Actions Taken metadata and tuples. Although the description of the recommendation engine in the context of Figure 4 is to provide a predefined set of actions in the form of recommended checklists, in the more general setting the recommendations could be new information sources that might be relevant for a given decision.

One of the challenges with regard to the development of a recommendation engine is how the system should "understand" and adapt to the various biases inherent in the way humans explore their information environment? For example, information bias (the tendency to seek information even when it cannot affect action), confirmation bias (the tendency to search for or interpret information or memories in a way that confirms one's preconceptions) and anchoring (the tendency to rely too heavily, or anchor, on one trait or piece of information when making decisions) may be guiding the humans information seeking patterns. Any recommender system, through its ability to better manage and understand user-context and the decision making environment, should help overcome these limitations.

X. WORKFLOW MANAGER

Although the specific example of a 24x7 Watch Floor is used to describe the concepts of APTO, the intent of the architecture is to accommodate a broader class of problems. The general characteristics of these problems are that they have a high volume of streaming and static data that is composed of structured components and unstructured data (predominately text data). The unstructured data can be given structure in the form of an event assertion (a semantic tuple). From the combination of the original structured components and the discovered event assertions, events can be determined. Once an event (or set of events) is determined, a set of actions needed to respond to the event can be determined. In many, but not all, situations, it is desired that the system identify, track and remember the actions taken.

Depending upon the specific domain or scenario being addressed by APTO, only some of these process steps are required to reach the objective of having actionable information upon which to make a decision. To accommodate different workflows (or process steps), the APTO architecture is comprised of independent, re-usable modules whose interactions represent a workflow. Every module in the architecture reports what it has done to the Workflow Manager. For example, when a new event assertion is created, the Workflow Manager is notified. Based upon the notification received and the specific workflow that is being executed, the next process step is determined and executed. It is envisioned that there may be multiple concurrent workflows executing within APTO.

A. Domain Specific Workflow

A Domain Specific Workflow component defines how data (objects) flow through the APTO architecture, determines which Action Taken items are important, and which Action Taken items trigger new Activities (or Action Sets).

B. Actions Taken

The Actions Taken component contains all of the actions that have occurred within the APTO architecture. Similar to our Target Events and Reports, the Action Taken domain object is a collection of metadata and a list of assertions (tuples). Essentially, an Action Taken item is a realized instance of an action semantic model. Where the model in the Prescriptive Action Set identifies what "should" occur, the Action Taken object identifies what actually happened answering the "Who", "What", "When", "Where", and "Why" questions.

CONCLUSION

This paper has discussed a context aware Proactive Decision Support framework within the OZONE environment. Furthermore, several longer term challenges have been briefly described with regard to modeling decision context, metrics for recognizing operational context, and techniques for recognizing context shift. Additional research areas include:

- Adequately capturing users' information interaction (seeking) patterns (and subsequently user information biases)
- Reasoning about information seeking behaviors in order to infer decision making context; for example, the work being done by researchers within the Contextualized Attention Metadata community [11] and the Universal Interaction Context Ontology [12] might serve as a foundation
- Instantiating formal models of decision making based on information seeking behaviors
- Leveraging research from the AI community in plan recognition to infer which decision context (model) is active, and which decision model should be active
- Recognizing decision shift based on work that has been done in the Machine Learning community with "concept drift", and assessing how well this approach adapts to noisy data and learns over time
- Incorporating uncertainty and confidence metrics when fusing information and estimating information value in relation to decision utility

Elaborating further on the ideas presented in the paper, longer term research should be focused on the following:

Decision Models for goal-directed behavior: Instantiation of prescriptive models of decision making, which integrate information recommendation engines that are context-aware. Furthermore, techniques that can broker across, generalize, or aggregate, individual decision models would enable application in broader contexts such as group behavior. Supporting areas of research may include similarity metrics that enable the selection of the appropriate decision model for a given situation, and intuitive decision model visualizations.

Information Extraction and Valuation: Locating, assessing, and enabling, through utility-based exploitation, the integration of high-value information within the decision models, particularly in the big data realm is a research challenge due to the heterogeneous data environment. In addition, techniques that can effectively stage relevant information along the decision trajectory (while representing, reducing and/or conveying information uncertainty) would enable the wealth of unstructured data to be maximally harnessed.

Decision Assessment: Modeling decision "normalcy", in order to identify decision trajectories that might be considered outliers and detrimental to achieving successful outcomes in a given mission context would be areas for additional research. Furthermore, techniques that proactively induce the correct decision trajectory to achieve mission success are also necessary. Lastly, metrics for quantifying decision normalcy in a given context can be used to propose alternate sequences of decisions or induce the exact sequence of decisions. This would require the pre-staging of the appropriate information needed to support the evaluation of those decisions and would potentially improve the speed and accuracy of decision making.

Operator/Human Issues: Understanding, modeling and integrating the human decision making component as an integral part of the aforementioned areas is a novel areas of research. The challenges are to represent human decision-making behavior computationally, to mathematically capture the human assessment of information value, risk, uncertainty, prioritization, projection and insight; and computationally representing human foresight and intent.

REFERENCES

[1] Filip Radlinski, Martin Szummer, Nick Craswell. Inferring query intent from reformulations and clicks. • Proceedings of the 19th international conference on World wide web. Pages 1171-1172. ACM New York, NY.

[2] Mohr, P. H., Ryan, N., & Timmis, J. (2006). Capturing Regular Human Activity through a Learning Context Memory. In *Proceedings of the 3rd International Workshop of Modelling and Retrieval of Context (MRC 2006) in conjunction with AAAI-06* (p. 6).

[3] Mohr, P.; Timmis, J.; and Ryan, N. (2005), Immune inspired context memory. In *1st International Workshop on Exploiting Context Histories in Smart Environments*, 4.

[4] Neal, M. (2003), *Meta-stable Memory in an Artificial Immune Network, Proceedings of the 2nd International e-Conference on Artificial Immune Systems*, p. 229-241.

[5] Agrawal, Vikas, Heredero, Genoveva, Penmetsa, Harsha, Laha, Arijit, and Shastri, Lokendra. "Activity Context Aware Digital Workspaces and Consumer Playspaces: Manifesto and Architecture" AAAI Workshops (2012): n. pag. Web. 14 Aug. 2013

[6] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction from the web. In IJCAI, pages 2670-2676, 2007.

[7] Ramesh Jain: EventWeb: Developing a Human-Centered Computing System. IEEE Computer 41(2): 42-50 (2008)

[8] Jim Hendler: Web 3.0 Emerging. IEEE Computer 42(1): 111-113 (2009).

[9] Anthony Fader, Stephen Soderland, Oren Etzioni: Identifying Relations for Open Information Extraction. EMNLP 2011: 1535-1545.

[10] Utz Westermann, Ramesh Jain: Toward a Common Event Model for Multimedia Applications. IEEE MultiMedia 14(1): 19-29 (2007).

[11] http://www.dlib.org/dlib/september07/wolpers/09wolpers.html, retrieved on 1 November 2013.

[12] A. Rath, D. Devaurs, and S. Lindstaedt. UICO: an ontology-based user interaction context model for automatic task detection on the computer desktop. In CIAO '09: Proceedings of the 1st Workshop on Context, Information and Ontologies, page 10, NewYork, NY, USA, 2009. ACM

Dynamic Data Relevance Estimation by Exploring Models (D^2REEM)

H. Van Dyke Parunak
Soar Technology, Inc.
3600 Green Court, Suite 600
Ann Arbor, MI 48105 USA
van.parunak@soartech.com

***Abstract*—Analysts in many areas of national security face a massive (high volume), dynamically changing (high velocity) flood of possibly relevant information. Identifying reasonable suspects confronts a tension between data that is too atomic to be diagnostic and knowledge that is too complex to guide search. D^2REEM (Dynamic Data Relevance Estimation by Exploring Models) is a knowledge-based metaheuristic that uses stochastic search of a graph-based semantic model to guide successive queries of high-volume, high-velocity data. We motivate D^2REEM by considering the nature of knowledge-based search in high-volume, high-velocity data and reviewing current tools. We then outline the D^2REEM metaheuristic and describe the state of progress in applying it to a range of model types, including geospatial movement, behavioral models, discourse models, narrative generators, and social networks. Finally, we outline work that needs to be done to advance the D^2REEM agenda.**

Keywords—retrieval, querying, semantic models, big data, stochastic search, any-time methods

I. Introduction

Analysts in many areas of national security face a massive, dynamically changing flood of possibly relevant information. "Big data" is typically described in terms of Volume (the amount of data), Velocity (how fast it changes), and Variety (the diversity of data formats); our concern here focuses on high-volume, high-velocity data. Activities of crucial interest can be expected to leave many "footprints" in available data, but identifying reasonable suspects confronts a tension between *data that is too atomic* to be diagnostic and *knowledge that is too complex* to guide search.

The *data* problem is that no single data item is diagnostic of an attack. Any one data item that might be part of an attack could also be part of a benign scenario. For example, a purchase of fermentation equipment might be a precursor to anthrax cultivation...or to opening a microbrewery. A new dissertation on gene splicing in microbes might point to a potential perpetrator...or just a promising new assistant professor. In data retrieval terms, static single-item queries give very low precision in identifying the overall event.

The *knowledge* problem is that while we can capture overall patterns of behavior that are diagnostic, matching them against massive data is combinatorially prohibitive. Representations that are available include discourse models that capture the different forms a conversation in social media might take [1, 2], hierarchical task networks (HTN) that capture goal-oriented behaviors [3, 4], social networks that show possible connections and flows among people and organizations [5, 6], and narrative models that capture causal dependencies [7]. Such a structure covers many possible behaviors, depending on which combinations of constraints are satisfied. If we could match such a structure against data, we would expect very high precision and recall. However, realistic structures can grow very large (for instance, an HTN might contain hundreds or thousands of atomic behaviors and constraints), and naïvely matching such a structure against massive data all at once is combinatorially prohibitive.

This paper describes D^2REEM (Dynamic Data Relevance Estimation by Exploring Models), a knowledge-based metaheuristic that uses stochastic search of a semantic model to guide successive queries of high-volume, high-velocity data. Section II explores the challenge that D^2REEM addresses and the current state of the art. Section III outlines the D^2REEM metaheuristic. The heart of D^2REEM is a knowledge-based model of the domain, and Section IV reviews several classes of models to which D^2REEM may be applied and documents our success so far. D^2REEM is a work in progress. Section V identifies a series of next steps for advancing this approach to semantic-driven search of big data. Section VI concludes.

II. The Challenge and Prior Work

Figure 1 summarizes the challenge that D^2REEM addresses. Static single-record queries are simple, but can be efficiently applied to high-volume high-velocity data. Conventional matching methods are too inefficient to apply knowledge-rich patterns to such data. D^2REEM is a novel way to match complex patterns to big data.

For years, the staple of information retrieval has been the record-oriented query, in which the analyst describes single data items that might be of interest. Static queries can be matched very efficiently, but their relevance depends on the state of knowledge about the world, which changes with each new piece of information.

The last 25 years have produced an explosion in graph databases, that is, databases that capture semantic relationships among data items in a graph structure. Graph databases can be

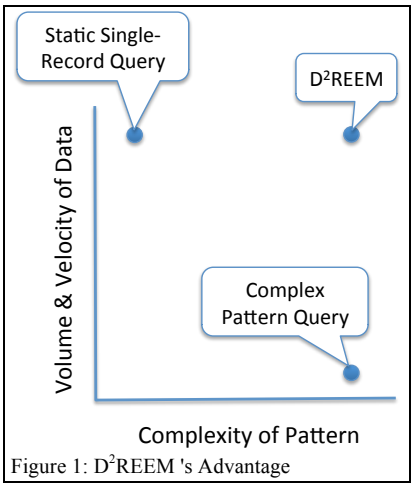


Figure 1: D²REEM 's Advantage

used to answer a range of queries in such data, including subgraph matching (does a specified pattern appear), shortest path discovery, path comparison, and computation of aggregate graph properties. Our focus in this paper is on subgraph matching. Queries against graph databases are done by specifying constraints over multiple nodes, such as a subgraph of the database, or a path that satisfies certain criteria, or specified aggregate characteristics of the graph [8]. Examples of such query languages are Cypher for Neo4J [9], or XPath for XML [10], or SPARQL [11] for RDF [12].

Graphs are a natural way to capture a knowledge model, but classical graphical query languages have several disadvantages for knowledge-based subgraph matching.

- They are generic to any graph-structured data, and do not take advantage of specific semantics in various kinds of graphical models. We wish to exploit the knowledge in a model.
- They require the entire query to match a subset of the data. We would like to search the data against a graphical structure (such as a hierarchical task network [HTN]) that expresses a range of possibilities, and identify coherent subsets of the pattern that the data support.
- In general, graph matching is intractable [13], with either exponential or NP-complete complexity in the size of the query. Thus queries must be kept small [8]. We wish to exploit large knowledge models.
- Graph databases require the data to be represented as a graph. We address high-volume high-velocity data streams (such as social media) where such preprocessing is not feasible.

III. THE D²REEM METAHEURISTIC

D²REEM is a metaheuristic, a high-level procedure that guides a lower-level process (in this case, record-level querying). Like many metaheuristics (e.g., genetic algorithms, ant-colony optimization, swarm optmization, artificial immune systems), its methods are strongly inspired by biological models.

TABLE I. COMPARISON OF D²REEM WITH SUBGRAPH MATCHING IN GRAPH DBS

	Graph DB	D²REEM
Query Size	Small Expresses complete structure of interest Search is for the entire query graph	Large Describes a range of structures of interest Search is for a matching subset
Data	Graph-structured	Record-structured
Query Semantics	Implicit Depends on use of same graph grammar for query and data	Explicit Enforced by PSE and EPM
Matching	Match entire query graph against data	Repeatedly match most relevant query node against data
Processing	Focus is on matching query graph against data graph Complexity is NP complete (subgraph isomorphism)	Focus is on exploring query graph in light of current data, and pursuing information on most relevant node Complexity is linear in size of knowledge model

In this section we introduce the metaheuristic, then explore two of its key components in more detail. The next section discusses classes of knowledge models to which it can be applied, and surveys our experience so far with each of them.

A. Overview

D²REEM shifts the focus of computation in doing knowledge-based exploration of big data. It moves computation away from matching the model against the data, and toward executing a process over the model that embodies the distinctive semantics of the model. Table I summarizes the differences between D²REEM and subgraph matching in a graph database.

Because D²REEM works with data as a stream of records, rather than a pre-processed graph, it must issue many record-level queries in order to match a knowledge model. It does this by executing a continuous cycle (Figure 2). Repeatedly, D²REEM

- explores the current state of the model,
- updates the priority for learning more about each node in the model,
- adaptively generates a query for the highest-priority node, and
- updates the model with what is learned from that query.

The queries can be posed to any data source, and do not

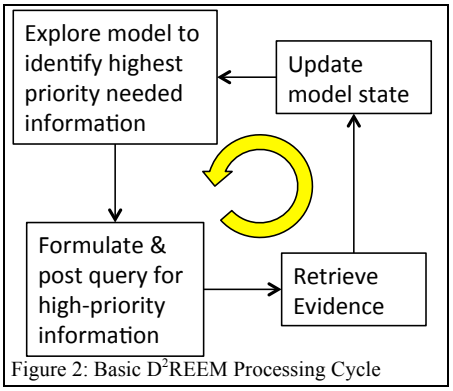


Figure 2: Basic D²REEM Processing Cycle

require predefining relationships among separate data items. The relationships among retrieved nodes are computed by exploring the model, not by a complex matching process, a strategy similar to graph simulation [14] (though unlike that work, we do not require that the data already form a graph).

Figure 1 shows the result. Static single-record queries can be applied to big data, but cannot capture complex patterns among records. Graphical databases can express patterns, but computational complexity forces the patterns to be smaller than a realistic behavioral model, and the data must be small enough and stationary enough to preprocess into a graph. By taking advantage of model semantics, D^{2}REEM can match very large knowledge-rich patterns (with thousands of nodes) against high-volume, high-velocity data streams that are not in graphical form.

Figure 3 shows the D^{2}REEM architecture. The heart of D^{2}REEM is a Graphical Knowledge Model (GKM) with two characteristics:

- Edges in the graph represent causal or other sequential dependencies between nodes, so that a trajectory is a possible evolution of the world, and
- The likelihood of visiting a given node can be modulated by evidence attached to the node.

The Polyagent Sampling Engine (PSE) continuously samples alternative trajectories through the GKM to generate a distribution over possible trajectories reflecting current knowledge of the domain. The Evidence Prioritizer and Marshaller (EPM) examines these distributions to identify nodes about which more information would be useful, issues queries to retrieve that information, and updates the GKM with the results. The PSE's ongoing exploration takes account of this new information, modifying the distributions over trajectories, and thus leading to new rounds of queries, implementing the processing cycle shown in Figure 2.

B. *Polyagent Sampling Engine*

By construction, each trajectory through a GKM corresponds to a possible instance of the dynamics implicit in the graph. Evidence currently on each node of the graph modulates the probability assigned to trajectories involving that node. We wish to construct a distribution over all possible trajectories. An approach we have found particularly tractable over many types of GKM is polyagent sampling.

A polyagent is a set of agents that collectively explore possible trajectories for a single entity or behavioral instance of interest. It consists of a single persistent coordinating agent (the "avatar"), which continuously generates a stream of simple agents ("ghosts"), each exploring a single trajectory. Figure 4 shows a polyagent sampling possible paths through a geospatial domain.

Ghosts have three biologically-inspired characteristics: they are manifold, apoptotic, and tropistic. "Manifold" means that many of them explore the domain in parallel, like multiple ants in an ant, or multiple chromosomes in genetic evolution, or multiple antibodies in an immune system, or agents in swarm optimization. "Apoptotic" means that they die after a fixed number of cycles. Thus the avatar can continue to generate new ghosts without overloading the system. "Tropistic" means that they move based on the characteristics of their environment, like ants. Physical ants plan paths through complex environments by depositing and responding to chemical pheromones. Polyagent ghosts respond to "digital pheromones," scalar fields maintained on the nodes of the GKM. These fields may reflect evidence supporting or refuting a given node. In addition, ghosts deposit a presence pheromone on each node that they visit. The normalized presence pheromone over the entire graph gives a probability distribution over possible trajectories of the entity that the polyagent represents.

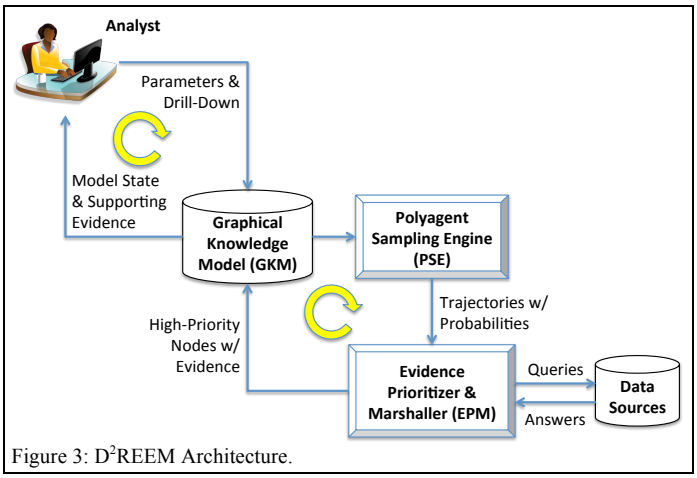


Figure 3: D^{2}REEM Architecture.

While the immediate inspiration of the PSE is biological, its mathematical underpinnings are based on Monte-Carlo tree search (MCTS) [15, 16], which explores multiple descendants of a single node to estimate the probability with which that node should be visited. In MCTS, the graph being explored is a game tree, in which the same game rules are applied in expanding each node. The PSE generalizes this concept to other graph structures, taking advantage of their distinctive semantics in the decision rules used by the ghosts and the digital pheromone fields they manipulate.

C. *Evidence Prioritizer and Marshaller*

The EPM has three functions:

- Based on the distribution of trajectories through the GKM determined by the PSE, identify the nodes for which additional information would be most valuable.
- Formulate and execute queries that will provide more information on the selected nodes

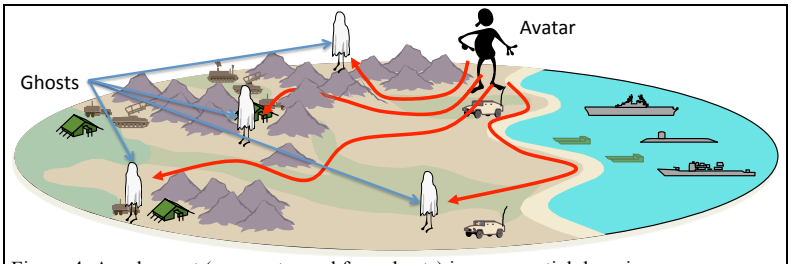


Figure 4: A polyagent (one avatar and four ghosts) in a geospatial domain

- Update evidence on the selected nodes based on the results of the queries.

We consider these in turn.

Identify nodes to guide queries.—Intuitively, D^{2}REEM estimates the relevance of candidate queries based on the nodes for which additional information would be most valuable. The precise sense of "valuable" depends on the kind of GKM that is guiding the search, and the decisions that it is guiding. Here are some alternatives that are useful in different settings. In the next section, we give further examples of each of these.

In sparse environments, the most valuable query is one *most likely to yield a hit*. In our PROPS system, polyagent sampling over a geospatial lattice generates candidate trajectories for adversaries, and the most probable trajectory guides the decision of where to deploy scarce surveillance assets to increase the probability of detecting an adversary.

One might try to *maximize some global measure* over the GKM. One use for an HTN in D^{2}REEM is to model a potential adversary's behavior (e.g., mounting a biological attack). In an HTN (using the rTÆMS dialect [4]), each leaf task that is executed contributes to the quality that accumulates at the root, and the higher that quality, the better the objective is achieved. By examining a set of possible trajectories identified by the PSE, the EPM can identify which trajectory would yield the highest root quality. If the HTN models adversarial behavior, this trajectory is most consistent with the adversarial intent we are seeking to detect. In this case, we want to select the nodes for which gaining more information might increase the probability of that trajectory.

In some cases, the nodes about which we want to learn more are those for which more information would *sharpen the distribution over alternative trajectories*. We estimate the effect of this choice by changing the evidence levels for various nodes in copies of the GKM and run the PSE on them, then compare the resulting distributions.

Formulate and Execute Queries.—The EPM submits queries to external data sources for those nodes that have been identified as of highest priority. Currently, we construct queries for each node manually in the course of formulating the GKM, and the EPM retrieves the specified query and submits it.

Update Node Evidence.—The EPM updates the evidence supporting the node on the basis of the response to the query. This change modulates the ongoing execution of the PSE, potentially changing the highest priority nodes in the next invocation of the EPM and directing the search process to the most relevant potential data.

IV. Examples of D^{2}REEM Models

The heart of D^{2}REEM is a semantic model of some facet of the real world. We have identified numerous such models, and demonstrated various facets of D^{2}REEM on them. This section outlines these examples. For each, it discusses

- how the model supports the two requirements identified in Section III.A (trajectories represent possible evolutions of the world; evidence on nodes modulates probability of trajectory)
- how it supports the PSE and EPM (in particular, what makes a node "relevant"), and
- what aspects of D^{2}REEM have been implemented in it.

A. Movement on Geospatial Maps

The most mature class of GKM for polyagent sampling is the geospatial lattice, whose nodes correspond to tiles of the environment and whose edges represent adjacency among tiles [17].

A trajectory represents the movement of a target, and the probability that a trajectory visits a node depends on externally-provided information such as terrain, presence of friendly or adversarial forces, and combat activity. The cumulative distribution of presence pheromone thus reflects the probability of encountering the target as a function of location.

In the DARPA RAID project, we applied the PSE on such a model to urban combat. Figure 5 shows the close correlation of predictions of red force movement in a human-commanded wargame, compared with the actual movement of troops. Quantitatively, the PSE produced more accurate forecasts than both experienced human observers and game-theoretic or Bayesian reasoners [18].

In the ARL PROPS project, we used the PSE on a geospatial lattice to direct collection management. The relevance criterion in this case is to give priority to queries (intelligence requests) on areas most likely to generate a hit.

PROPS is the most mature implementation of the D^{2}REEM metaheuristic to date, including ongoing PSE exploration of the knowledge model, dynamic query formulation, and updating of the knowledge model.

B. Hierarchical Task Networks

Goal-oriented behavior by intelligent agents is often represented with a hierarchical task network (HTN) [4, 19]. Figure 6 is a fragment of an HTN model for a mix of benign and nefarious cyber-activities. The nodes are actions, and are joined by two kinds of edges: subtask edges (solid) that connect a higher-level task (the goal) to lower-level tasks that

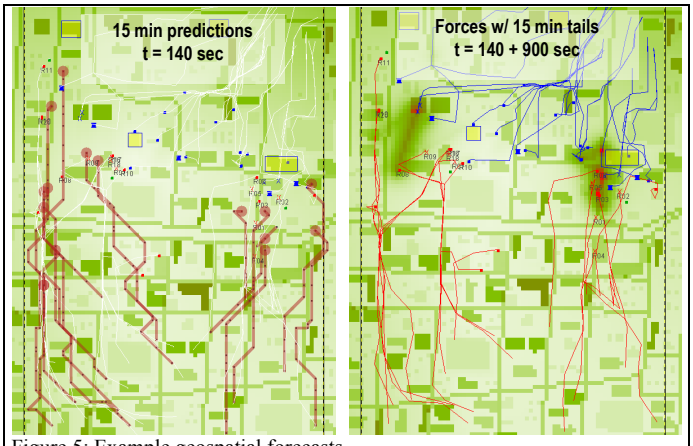


Figure 5: Example geospatial forecasts

carry it out, and sequence edges (dashed) that reflect precedence constraints. These precedence constraints are inherited by the leaves of the HTN. The graphical language in the figure is a simplification; our full formalism, derived from the TÆMS language [19], is much more sophisticated. In TÆMS, successful execution of a task generates "quality," a scalar value, that propagates upward via combination rules. The degree to which a sequence of leaf actions satisfies a top-level goal is measured by the amount of quality that accumulates at that top node.

Polyagent sampling explores alternative trajectories through the leaf tasks. Each trajectory reflects a sequence of actions that an agent might execute in the world. The probability that an agent's next step will move to a given task depends on the task's feasibility (the satisfaction of its prerequisites), its desirability (based on the degree to which higher-level tasks have been achieved), and evidence for the task from the external world (provided by the EPM).

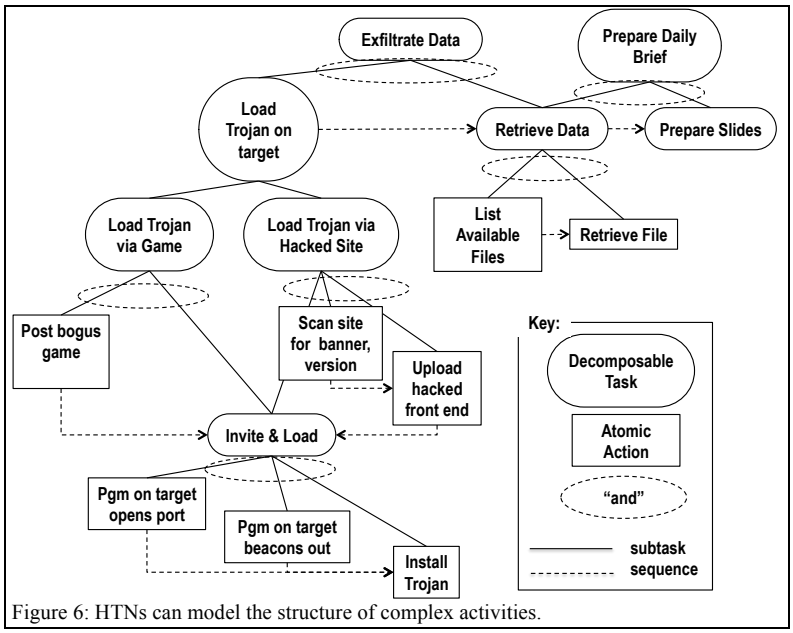


Figure 6: HTNs can model the structure of complex activities.

The HTN is an example of a GKM for which the value of generating a query for a node depends on a global characteristic of the model, namely, the change in the quality of the root node that a response to the query might generate. We have demonstrated the PSE on HTNs [4, 20], but not yet implemented an EPM for it.

C. Social Networks

We represent a social network [5] as a bipartite graph, in which one set of nodes represents people or organizations, and another indicates class of transaction. Several different kinds of transaction are possible, including communication, transfer of wealth, transfer of power (e.g., by confrontation), or transfer of status (e.g., by endorsement). Figure 7 is an example social network in our PSTK system (Power Structure ToolKit), in which the Agents are people and the bar graphs between them represent the levels of the different transaction types (in this example, Political, Military, Economic, Social, from the PMESII ontology).

A trajectory in a social network indicates a sequential transfer of social capital. For example, one may seek evidence for a money laundering operation that moves a financial payment makes its way through a series of organizations. Evidence of a specific transaction increases the likelihood of a transition from one agent to another.

Our current PSTK system explores possible flows using specialized processes residing on each agent, not the PSE. We have not implemented a EPM for social networks. If one is seeking to identify sequences of transactions, the relevance of a node to generate a query is measured by the degree to which additional information on that node would sharpen the distribution over alternative trajectories. For example, a node that is shared by several emerging trajectories would not rank as high as one that is unique to a single candidate trajectory.

D. Narrative Space Models

A Narrative Space Model (NSM) captures a set of many possible narratives that could explain the evolution of a situation [7, 21], and is an external representation for the mental activity of an analyst who is seeking to explain how a given state of affairs might come about. Each node in an NSM consists of a statement about the world to which belief may be assigned. The NSM has an edge from one node to another just if the first statement followed by the second forms a coherent segment of narrative.

Each trajectory through an NSM represents a coherent narrative about how the world might evolve from the origin to the destination. Figure 8 shows an abbreviated NSM that captures ways that al-Assad might stay in power or lose power in Syria. The '????' notation on edges between nodes are placeholders for edge weights that the PSE fits based on evidence on the nodes. In an NSM, external evidence for (against) an individual node increases (decreases) the probability of trajectories including that node.

We have implemented the PSE on NSMs, and modulated its behavior by attaching external evidence to nodes in the

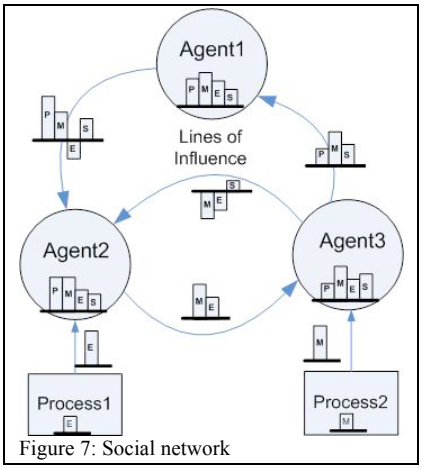


Figure 7: Social network

NSM. In our work so far, this evidence has been attached by a human analyst, not by the EPM. Since our interest is in identifying the most likely narrative given the evidence available to date, the EPM for a NSM would weight nodes based on how much evidence for a given node would sharpen the distribution over emerging narratives.

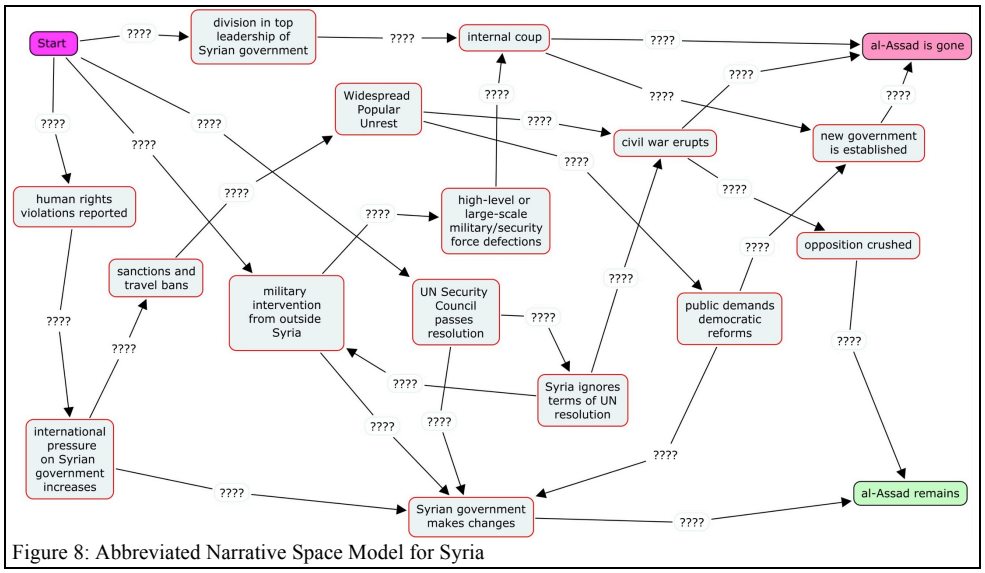


Figure 8: Abbreviated Narrative Space Model for Syria

E. *Discourse Models*

Dooley Graphs [2] reflect a speech-act view of discourse [22], in which each utterance seeks to accomplish something (e.g., Solicit an action or a statement, Inform, Commit, or Refuse). In a coherent conversation (a sequence of speech acts), later utterances may be related in different ways to earlier ones: they may Respond, Reply, Resolve, or Complete them. Detecting such coherent conversations from a high-volume, high-velocity stream of data (for example, a Twitter feed) would make a great contribution to surveillance activities.

There are a number of ways one could graph a sequence of utterances, depending on what one chooses as the nodes.

- The nodes could represent specific *utterances*, and edges would reflect the sequences among them. This representation loses critical information about who issues each utterance.
- One might analyze the conversation to characterize different states that it could assume, and then represent each *state* as a node, with edges representing possible state transitions. A state model, like an utterance model, deemphasizes the participants, and in addition makes it difficult to distinguish specific conversations.
- We could represent *participants* as nodes, with edges representing utterances from the source to the target. Like the state model, the participant model does not clearly capture the progress of an individual conversation.

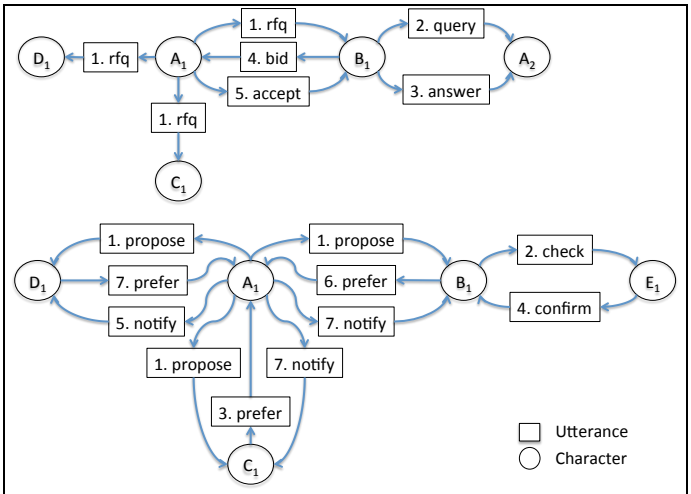


Figure 9: Example Dooley Graphs for purchase negotiation (top) and arranging a meeting (bottom)

A Dooley graph (e.g., Figure 9) is a bipartite graph. One class of nodes (circles in Figure 9) represents *characters*, which are participants at distinguished stages of the discourse, based on the notions of resolution and completion. Thus participant A may appear as nodes A_1 and A_2. The other class of nodes (squares in Figure 9) represents utterances, which are characterized by type of speech act. Utterances that resolve or complete one another tend to form tightly-connected components of the graph, while those that take off in new directions spawn new components. A trajectory through the Dooley graph represents a realization of a conversation. Retrieving a tweet from (say) A to B adds evidence to A-characters and B-characters; recognizing the tweet as a specific speech act adds evidence to utterance nodes requiring that speech act.

We have not yet implemented either the PSE or the EPM on Dooley Graphs. In using a Dooley Graph for surveillance of social media, one would seek to identify well-formed conversations and classify them (e.g., meeting organization, viral propagation of opinion, purchase activities). For this purpose, the EPM should prefer nodes based on their potential for sharpening the distribution over alternative trajectories.

V. NEXT STEPS

Four main avenues for extension of D^{2}REEM provide a range of challenging and important research opportunities: multiple model types, model management, query generation, and model linking.

Multiple Model Types: Our most complete example of D^{2}REEM is the PROPS system, which treats the geospatial domain. The NSM is the next most mature, demonstrating the effectiveness and computational efficiency of modulating the state of a non-geospatial knowledge model by external evidence. In addition to refining these applications, we seek to extend the complete D^{2}REEM cycle to other model types. As we configure the PSE and EPM to different model types, we

gain valuable insights into how the underlying mechanisms of the metaheuristic can be generalized.

Model Management: As noted in Section II.A, an important difference between D^{2}REEM and graph DBs is how knowledge is expressed. Graph DBs construct a small graphical query using the same graph syntax that governs the data, and seek to match the entire query graph against the data graph. D^{2}REEM uses a large GKM that captures a range of hypotheses, and then explores this model in the light of the data to identify high-priority record-level queries. The use of a complex knowledge model is a strength, in that it externalizes analysts' internal mental models, facilitating collaborative review and enhancement. But it is also a weakness, since constructing such models is itself a labor-intensive process.

For many long-standing problems, model construction is integral to the analytic effort [21], and D^{2}REEM offers an additional incentive to construct such models. But it will be even more useful if model construction can be partially automated. For example, in the case of the NSM, techniques exist to merge specific narratives in a domain of interest into a NSM [23, 24]. Such technology could exploit past analytic products (which often include a narrative of the event under investigation) to enhance a NSM of the domain. Another example is the Disciple technology [25], which has been used successfully to learn inferential relations of the sort one might encounter in a belief network.

A strength of the PSE approach to model exploration is the locality of ghost movement and pheromone-based interaction. This locality means that GKMs can be extended incrementally, and encourages the notion of a persistent library of dynamically updated models as a central resource in analysis. Development of mechanisms for managing such a library will considerably advance the analytic enterprise.

Query Generation: One task of the EPM is formulating queries that can provide additonal information on GKM nodes that it identifies as highly relevant. In our current implementations, these queries are manually constructed along with the GKM. Given the description of a node in a model and schemata for external data sources, one would like to generate queries automatically, a task that will rely heavily on research in ontological reasoning and semantic web technologies.

Model Linking: The same analytic tasking can be viewed through the lens of multiple model types, and we would like to facilitate the flow of information between these model types by defining mappings between nodes in different model types. Like the previous topic, this one depends on advances in ontological reasoning, as well as model theory and other formal tools [26], and will require attention to aligning multiple levels of meaning [26, 27], not all of which may be represented in each model.

VI. CONCLUSION

Matching knowledge-rich patterns against high-volume, high-velocity data is combinatorially prohibitive. The D^{2}REEM metaheuristic is a new approach to such retrieval problems that shifts the computational burden from graph matching (a NP-complete problem) to iteratively exploring a knowledge model and issuing focused queries for the data that is most relevant in the light of current knowledge (a process that is linear in the size of the knowledge model). D^{2}REEM can be applied to any graphical knowledge model in which edges represent causal or or other sequential dependencies and in which adding data to individual nodes can change the probability of a trajectory.

REFERENCES

[1] N. N. Binti Abdullah, "Activity States: a theoretical framework for the analysis of actual human collaboration on the Web," Ph.D., Information, Structure, Systèmes, Université Montpellier II, Montpellier, France, 2006.

[2] H. V. D. Parunak, "Visualizing Agent Conversations: Using Enhanced Dooley Graphs for Agent Design and Analysis," in *Second International Conference on Multi-Agent Systems (ICMAS'96)*, 1996, pp. 275-282.

[3] K. Erol, "Hierarchical Task-Network Planning Systems: Formalization, Analysis, and Implementation," Computer Science, University of Maryland, College Park, MD, 1995.

[4] H. V. D. Parunak, T. Belding, R. Bisson, S. Brueckner, E. Downs, R. Hilscher, and K. Decker, "Stigmergic Modeling of Hierarchical Task Networks," in *the Tenth International Workshop on Multi-Agent-Based Simulation (MABS 2009, at AAMAS 2009)*, Budapest, Hungary, 2009, pp. 98-109.

[5] G. Taylor, B. Bechtel, K. Knudsen, E. Waltz, and J. White, "PSTK: A Toolkit for Modeling Dynamic Power Structures," in *the 16th Behavior Representation in Modeling and Simulation Conference (BRIMS)*, Providence, RI, 2008.

[6] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge, UK: Cambridge University Press, 2010.

[7] H. V. D. Parunak, S. Brueckner, L. Downs, and L. Sappelsa, "Swarming Estimation of Realistic Mental Models," in *Thirteenth Workshop on Multi-Agent Based Simulation (MABS 2012, at AAMAS 2012)*, Valencia, Spain, 2012, pp. 43-55.

[8] P. T. Wood, "Query Languages for Graph Databases," *SIGMOD Record,* vol. 41, pp. 50-60, March 2012.

[9] Neo Technology, *The Neo4j Manual*, 1.8.1 ed.: Neo4j, 2012.

[10] A. Berglund, S. Boag, D. Chamberlin, M. F. Fernández, M. Kay, J. Robie, and J. Siméon. (2011, 22 August). *XML Path Language (XPath) 2.0 (Second Edition)*. Available: http://www.w3.org/TR/xpath20/

[11] S. Harris and A. Seaborne, "SPARQL 1.1 Query Language, W3C Working Draft," W3C2011.

[12] G. Klyne and J. J. Carroll, "Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation," W3C2004.

[13] P. Barcelo, L. Libkin, and J. Reutter, "Querying Graph Patterns," presented at the PODS'11, Athens, Greece, 2011.

[14] M. R. Henzinger, T. A. Henzinger, and P. W. Kopke, "Computing Simulations on Finite and Infinite Graphs," presented at the the □□36th Annual IEEE Symposium on Foun□dations of Computer Science (FOCS 95), Milwaukee, WI, 1995.

[15] M. Kearns, Y. Mansour, and A. Ng, "A Sparse Sampling Algorithm for Near-Optimal Planning in Large Markov Decision Processes," presented at the the Sixteenth International Joint Conference on Artificial Intelligence, 1999.

[16]L. Kocsis and C. Szepesvári, "Bandit based Monte-Carlo Planning," presented at the the EMCL 2006, Berlin, Germany, 2006.

[17]H. V. D. Parunak, S. A. Brueckner, R. Matthews, and J. Sauter, "Swarming methods for geospatial reasoning," *International Journal of Geographical Information Science,* vol. 20, pp. 945-964, 2006.

[18]H. V. D. Parunak, "Real-Time Agent Characterization and Prediction," presented at the International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'07), Industrial Track, Honolulu, Hawaii, 2007.

[19]B. Horling, V. Lesser, R. Vincent, T. Wagner, A. Raja, S. Zhang, K. Decker, and A. Garvey. (2004, 1 August). *The Taems White Paper* [Web site]. Available: http://dis.cs.umass.edu/research/taems/white/

[20]S. Brueckner, T. Belding, R. Bisson, E. Downs, and H. V. D. Parunak, "Swarming Polyagents Executing Hierarchical Task Networks," in *Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2009)*, San Francisco, CA, 2009, pp. 51-60.

[21]L. Sappelsa, H. V. D. Parunak, and S. Brueckner, "The Generic Narrative Space Model as an Intelligence Analysis Tool," *American Intelligence Journal,* vol. 31, p. (in press), 2013.

[22]P. R. Cohen and C. R. Perrault, "Elements of a Plan-Based Theory of Speech Acts," *Cognitive Science,* vol. 3, pp. 177-212, 1979.

[23]B. Li, S. Lee-Urban, D. S. Appling, and M. O. Riedl, "Crowdsourcing Narrative Intelligence," *Advances in Cognitive Systems,* vol. 1, pp. 1-18, 2012.

[24]M. O. Riedl and R. M. Young, "From Linear Story Generation to Branching Story Graphs," *IEEE Computer Graphics and Applications,* vol. 26, 2006.

[25]Gheorghe Tecuci, M. Boicu, C. Boicu, D. Marcu, B. Stanescu, and M. Barbulescu, "The Disciple-RKF Learning and Reasoning Agent," *Computational Intelligence,* vol. 21, pp. 462-479, November 2005.

[26]A. Tolk, S. Y. Diallo, and J. J. Padilla, "Semiotics, entropy, and interoperability of simulation systems: mathematical foundations of M&S standardization," presented at the Proceedings of the Winter Simulation Conference, Berlin, Germany, 2012.

[27]R. Ackoff, "From Data to Wisdom," *Journal of Applied Systems Analysis,* vol. 16, pp. 3-9, 1989.

Data Analytics to Detect Evolving Money Laundering

Murad Mehmet, Duminda Wijesekera
George Mason University
mmehmet@gmu.edu , dwijesek@gmu.edu

***Abstract*— Money laundering evolves using multiple layers of trade, multi trading methods and uses multiple components in order to evade detection and prevention techniques. Consequently, detecting money laundering requires an analytical framework that can handle large amounts of unstructured, semi-structured and transactional data that stream at transactional speeds to detect business-complexities, and discover deliberately concealed relationships. Based on our prior work and a static risk model proposed in the Bank Security Act, we propose a dynamic risk model that assigns a risk score for every transaction being a potential component of a larger money-laundering scheme. We use social networks to connect missing links in such potential transaction sequences. Taken together we can provide a financial sector independent risk assessment to submitted transactions. The proposed risk model is validated using data from realistic scenarios and our already developed money laundering evolution detection framework (MLEDF) that we developed earlier using sequence matching, case-based analysis, social networks, and complex event processing to link fraudulent transaction trails. MLEDF has components to collect data, run them against business rules and evolution models, run detection algorithms and use social network analysis to connect potential participants.**

Keyword: Data Analytics; Social network analysis; Anti Money Laundering; Dynamic Risk Model ; Money laundering Risk .

I. INTRODUCTION

Money laundering (ML) is a major issue for the Department of Homeland Security (DHS) and US Treasury. Powered by modern tools, money launderers use complex schemes to avoid being detected by anti-money laundering (AML) systems. They dynamically evolve, expand and contract over fraudster networks in different countries. Social Network Analysis (SNA) techniques [1] are used by government agencies to track terrorist activities and networks. Because terrorist financing heavily depends on ML [2], any AML system must incorporate SNA to obtain reliable results. Schwartz [3] proposes a model to find criminal networks using social network analysis, building upon Borgatti's SNA-based key player approach [3]. One drawback of Borgatti's model is the failure to assign weights to actors and actor-actor relationships. In the recent past, we have developed algorithms incorporates Borgatti's SNA techniques with different weights' to social and business relationships to help complete missing links in potential money laundering chains.

The Financial Action Task Force (FATF) provides a static risk assessment of ML [4] strategies to examine ML related predicate crimes and known weaknesses of anti-money laundering (AML) systems. The Wolfsberg Group, made up of eleven leading international banks established standards, guidelines and a discretionary risk model [5] to counter money laundering. Both FATF and Wolfsberg Group say that monitoring customers is an essential part of countering money laundering and suggest that risks can be measured using metrics such as "Country risk", "Customer risk", and "Services risk", and leave weights assigned to each of these categories at the discretion of the evaluating organization. Based on these guidelines, banks use a quantitative model to evaluate transactional risk using attributes such as "Customer profile", "Product/service profile", and "Geographic profile".

The static risk model developed by Scor [6] accepts ML to be determined by "Agility" of adopting new rules per customer, "Complexity" of transactions and the "Secrecy" of transactional information and customer account" [6], but fails to assess other factors such as relationship networks, and dynamically changing factors. Kount [7] developed a dynamic scoring service to continuously monitor indicators of fraudulent credit card activity, and alert merchants of approved transactions that are linked to suspicious purchasing activities, that usually occur after identity theft. These suspicious purchases refer are transactions patterns that have never been witnessed before, such as the purchase of video games by senior citizen.

The rest of the paper is organized as follows. Section 2 describes the Money Laundering Evolution Detection Framework (MLEDF). Section 3 describes the SNA algorithm. Section 4 describes the dynamic risk model. Section 5 evaluates the performance of the MLEDF and dynamic risk model using real-life cases. Section 6 concludes the paper.

II. ML EVOLUTION DETECTION FRAMEWORK (MLEDF)

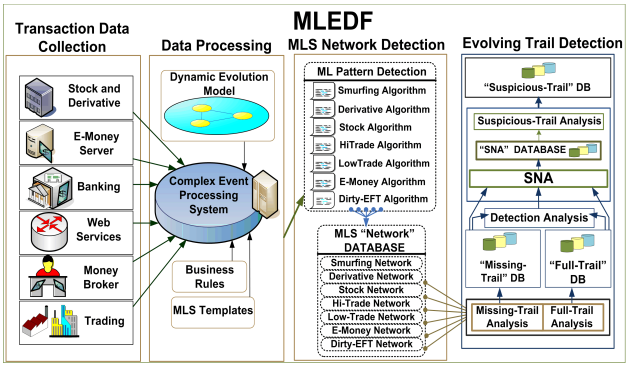


Fig. 1. Money Laundering Evolution Detection Framework

First we briefly describe how MLEDF works [8]. Obtaining data streams from multiple sources listed in the left hand column of Fig.1, using a complex event processing system. MLEDF uses four phases where output from one phase is used by the following phase and shown in the columns of Figure1.

(1) *Collecting Transactional Data*: Transaction processors or data collectors from Automated Clearing House such as EPN, FEDWIRE, and CHIPS send their data belonging to trade sectors such as Banking, Stock market, Derivative market, Web

based Services, Trading, Electronic Money, and Money Brokering. Relevant information is extracted form this data and used with transaction-independent data such as the economic status of the country, stock sales trends and the stock values during the day.

(2) *Data Processing*: Well-known MLS are identified and relevant attributes are collected from input data streams and submitted to our detection algorithms. The extracted data associated with each MLS pattern assigned to a specific MLS type using the following: (I) Business Rules: MLS business rules and red flags associated with each pattern, the rules associated with specific sector are used by the MLS detection algorithms to identify the MLS patterns. (II) MLS Template: Well-known MLS templates will be used during this phase. Currently, the templates have seven major pattern types with their different subtype combinations. This acts as a repository of known MLS. If a new form of MLS is discovered, then it will be added to this Database. (III) ML Evolution Model: Determines if the evolution of MLS is within the accepted trend of our model [8].

(3) *Detecting MLS Networks*: We use six algorithm (one for each) to detect Smurfing, Trade, Stock, Derivative, E-Money, and Dirty Electronic Funds Transfer (Dirty-EFT) schemes. Each algorithm uses a different method to capture the network associated with the specific type of MLS and in real-time output the discovered networks associated with the specific MLS pattern into a different database. Then, the discovered networks are reformatted and saved in a single database referred to as the "Network" Database to facilitate efficient analysis of the links among MLS networks.

(4) *ML Trail Analysis and Evolution Detection*: Four separate algorithms are run to find the "*Full-Trails*" [8], "*Missing-Trails*", and "*Suspicious-Trails*" of MLS networks. "*Full-Trail*" is a concatenated sequence of related schemes (MLS) act by itself to transfer money from one MLS to another until it reaches the final MLS, of which the orchestrator (i.e. the money launderer) is referred to as the "*EndBoss*". Similarly the orchestrator of the first scheme is referred to as the "*StartBoss*". "*Associates*" are other people involved in the sequence of fraud. "*Missing-Trail*" is a short Full-Trail that does not exceed have more than three related MLSs. A sample output from the Full Train Algorithm is given in Table 1, where a network ID (assigned by our detection algorithm), the duration of the laundering activity, if the money was withdrawn after the third transaction, the amount of money and the detected Start Boss and the detected End Boss are provided. We assume that the Missing-Trail is a premature Full-Trail with broken parts and missing links or evidence. A "*Suspicious-Trail*" is a combination of discovered Full-Trails and/or Missing-Trails constructed using algorithms that incorporate SNA and numerical analysis techniques. The algorithm "Detection Analysis" determines the evolution of the "*Full-Trail*"s such as the change to the number of involved associates, the changes to the cost of laundering, and changes to the laundering locations.

III. SOCIAL NETWORK ANALYSIS IN MLEDF

In many cases money launderers intentionally obfuscate the money trail either by hiding it (for example by increasing the number of transactions and reducing the transacted amount), or using an unreported method such as a Hawala [15] (an honor based exchange system without records). As a solution, we use a social network among money launderers to link MLS trails when evidence of linkage is missing among transactions.

A. Using Complex Event Processing in the SNA Module

The critical question of ML experts is "*How fast and how well can we relate the different events in this universe of detected MLS?*" Using the Complex Event Processing (CEP) system StreamBase, we developed an algorithm to create chains of related MLSs where social or professional relations are used to transfer a fund to the next MLS until it reaches the final destination where the *End Boss* withdraws the money. If we modeled all of the chains as a separately and link them we run into a scalability issue in associating the multitude of different events of various MLSs. As a solution, we model each detected MLS as an event, and have various patterns of events categorized under six different types of MLS. For example, Full-Trail algorithm outputs a trail by using the functionality of CEP of perceiving the MLSs as a set of events. Without the CEP the MLS should dissolve into the constituent transactions to be analyzed and linked with other transactions from another MLS, consumes processing time and resources. The CEP can link MLSs, perceived as events, using various criterions without the need to add more complex sub-algorithms for each criterion. That is, the Full-Trail connects the dots that exist, but it is harder and slower to connect them without CEP capabilities. Full-Trail captures the trail in cases where all evidence are available, whereas the Suspicious-Trail attempts to construct the path where some edges along the path is missing.

B. Integrating the "SNA" Module into MLEDF

The "*Suspicious-Trail*" module is used to detect components of an actual "Full-Trail" even if there is a missing piece of evidence. This module investigates all available trails (Full-Trail and Missing-Trail) by using our SNA DB that contains the weights of relationships among MLS participants in order to determine if two trails are related by considering some attributes such as the amount of funds involved, location, affinity of participants, time, and methods used for laundering. Hence, the "*Suspicious-Trail*" module uses the "*SNA*" module to produce a new trail containing two or more trails that are related based on SNA even when we have not captured a transaction joining them or any other evidence. The new trail is created after making a calculation based on (SNA) results of a possible relationship between two or more Full-Trails and Missing-Trails. The generated evolution patterns and strategies are collected into the "*Suspicious-Trail*" Database.

TABLE I. SAMPLE OUTPUT OF THE FULL-TRAIL (MIN 4 MLS)

Networks	TrailID	Duration	Withdraw	Amount	StartBoss	EndBoss
4, 91, 98 ,..	1232	76 Days	No	723,234	Boss956	Boss 153
24, 315, ..	1208	89 Days	No	890,165	Boss 103	Boss 827
405,783, ..	9724	19 Days	No	200,230	Boss 284	Boss 725

C. The Components and Output of the "SNA" Module

The architecture of the social network algorithm is shown in Figure 2. The SNA module generates and continuously updates two databases as outputs. The "*SuspectWeight*" database contains the weight of all relations detected in the

MLEDF and the "*Relations*" database containing the time and the record of all detected relations among pairs in MLEDF as shown in the bottom of Figure 2. The relations we capture are: (1) *UniqueFullTrailBosses* creating "StartBoss"–"EndBoss" pairs of "Full-Trail"s. (2) *UniqueFullTrailAssociates* creating "Asscoiate"-"Asscoiate" pairs of "Full-Trail"s. (3) *UniqueMissingTrailAssociates* createing "Asscoiate"-"Asscoiate" pairs of "Missing-Trail"s. (4) *SchemaBosses* creating Hashes for "StartBoss"-"EndBoss" relations. (5) *SchemaAssociateBoss* creating Hashes for all detected "Boss-Associate" relations. (6) *SchemaAssociate* creates Hashes for all detected "Associate-Associate" relationships. This hash represents the combinations of relationships among the associates of the same MLS, even if they do not interact/transact with each other directly. (7) *Family* creating "Family" relation between lineage-wise related pairs. (8) *Business* creating business-wise related pairs. Each such relationship is shown in Figure 2. We compute these relationships and assign weights to them as shown in Algorithm 1 describe in Table II.

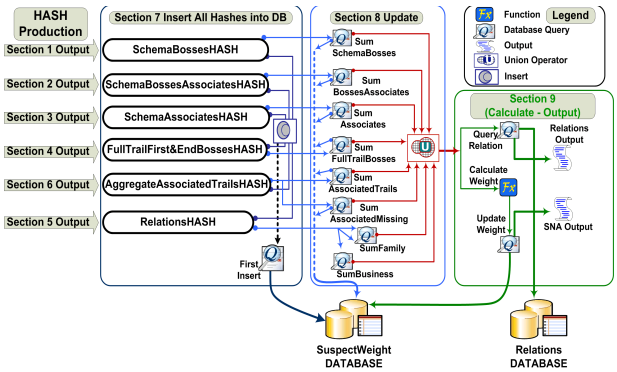


Figure 2: The Social Network Analysis Module

The relationship weights as assigned so that higher weight indicates more possible hidden interactions. Weights are calculated by adding parameters for each of the corresponding events as follows:

1. Each detected "MLS" weights of 10 will be added to start/end boss couple, 5 for each boss/associate combination, and 1 for each non-repeating associate/associate combination.
2. Each detected "Missing-Trail", 15 will be added to each associate non-repeating combination.
3. Each detected "Full-Trail" add 20 to each associate combination and 25 to the start and end boss.

Other strong relationships are also counted where "Family" ties will add 250 to the couple, and each "Business" relationship will add 250 to the couple. We chose the weights and, verified them in a limited engagement with a trusted third party (see Section V), but can be changed in Algorithm 1.

In the SNA Algorithm given in Table II, steps 1 and 2 define the hash function, and input and DBs constants associated with the different weights and the hash functions. Steps 3 and 4 create hashes for "*Boss-Boss*", "*Boss-Associate*", and "*Associate-Associate*" of MLSs. Steps 5 and 6 create the same hashes for Full-Trails. Steps 7 and 8 create hashes for Missing-Trails and special relations (of family and business). Step 9 computes the WeightOutput of a hash HRel. Sample Suspect Weights obtained from Algorithm 1 is shown in Table IV. This corresponds to Relationships given in Table III.

TABLE II. SOCIAL NETWORK ANALYSIS ALGORITHM

1	**FUNCTION** HASH (String1,String2){return concatenate(sort(En1,En2))};
2	**INPUT** MLS DetectedMLS; Relnship; MT MissingTrail; FT FullTrail; **DB HRel** (Hash, #"Time", Type, Person1, Person2) KEY (Hash, #"Time", Type); **DB SuspectWeightOutput** (hash, weight) KEY (hash);
3	**UPDATE** HRel SET **MLSBoss**++, **MLSAssocBoss**++, **MLSAssocBoss**++ WHERE HRel.hash == HASH(MLS.sBoss, MLS.eBoss) , HASH(MLS.Assoc, MLS.eBoss) , HASH(MLS.Assoc, MLS.sBoss) and TypeMatch
4	**FOR EACH** (MLS.Assoc as Assoc1, MLS.Assoc as Assoc2) **UPDATE** HRel SET **MLSAssoc**++ WHERE HRel.hash == HASH(MLS.Assoc1, MLS.Assoc2);
5	**FOR EACH** (FT.Assoc as Assoc1, FT.Assoc as Assoc2) **UPDATE** HRel SET **FTAssoc** = FTAssoc++ WHERE H.hash ==HASH(Assoc1,Assoc2);
6	**UPDATE** HRel SET **FTBoss++** WHERE HRel.hash == HASH(FT.sBoss, FT.eBoss);
7	**FOR EACH** (MTrail.Assoc as Assoc1, MTrail.Assoc as Assoc2) **UPDATE** HRel SET **MTAssoc++** WHERE HRel.hash == HASH(Assoc1, Assoc2) and TM;
8	**UPDATE** HRel SET **Business++, Family++** WHERE HRel.hash == HASH (Relnship.person1, Relnship.person2) AND Relnship.type == "BUSINESS","FAMILY";
9	**SELECT** HRel.hash, **(**25*HRel.FTBoss +20*HRel.FTAssoc +15*HRel.MTAssoc + 1*HRel.MLSAssoc + 5*HRel.MLSAssocBoss +10*HRel.MLSBoss + 250*H.Business +250*H. Family**)** as **SuspectWeightOutput** ;

TABLE III. SAMPLE SELECTION FROM OUTPUT OF "RELATIONS"

DetectTime	Hash	Type	Entity1	Entity2
2012121915	Comp10Comp8	FullTrailAssociates	Comp10	Comp8
2012121923	Comp10Comp5	SchemaBossAssociate	Comp10	Comp5
2012122005	Comp10Assoc7	MissingTrailAssociates	Comp10	Assoc7
2012122112	Assoc7Assoc5	SchemaAssociates	Assoc7	Assoc5
2012122214	Comp10EndBoss	SchemaBosses	Comp10	EndBoss
2012122220	StartBossEndBoss	FullTrailBosses	StartBoss	EndBoss

TABLE IV. SAMPLE SELECTION FROM OUTPUT OF "SUSPECTWEIGHT"

Weight	Hash	Weight	Hash	Weight	Hash
30	Comp10Comp8	10	Comp11Comp2	10	Assoc1Comp1
30	Comp10Comp7	10	Comp11EndBoss	30	Assoc1Comp4
15	Comp10Comp5	10	Comp1Comp2	35	Comp4Comp6
30	Comp10EndBoss	20	Comp6EndBoss	20	Assoc1Assoc5
10	StartBossEndBoss	20	Comp7Comp8	0	Assoc1Assoc9

IV. THE DYNAMIC RISK MODEL

Existing AML systems do not relate different products types, entities, and business lines involved in different combinations of complicated ML schemes. Industry specific AML systems use industry specific static risk models and therefore do not capture known dynamics of MLS evolutions. Countering ML and other forms of fraud requires industry-wide risk analysis method to where the risk score is updated dynamically and include transactional behavior related to the ML, such as the social relations and past associations with money laundering. Therefore, we create a dynamic risk model that incorporates the static attributes used by others, such as the senders and recipient's static profiles and dynamic social connection attributes of the transactions that we capture in our MLEDF system.

A. The Static Risk Model of Bnak Secrecy Act

Figure 3. The enhanced BSA Static Risk Modeling [9]

The Currency and Financial Transactions Reporting Act (CFTRA) of 1970 later amended to counter money laundering and financial crimes [11, 12, 13] and again amended by Title III of the PATRIOT Act of 2001 and other legislations, and is now commonly referred to as the "Bank Security Act" (BSA) mandates banks to monitor transactions and maintain records of initial and periodic risk scores for customers. Their risk model identify and analyze specific "products and services", "customers and entities", and "geographical locations" and categorize them as "high", "medium", and "low", and add the risk rating of all categories to obtain the overall accumulative risk score. We enhanced the BSA inspired static risk with aggregated static risk to reflect changing dynamics of ML and its consequences on the static risk calculation shown in Figure 3. The risk rates assigned in Figure 3 are obtained from [9], with suggested enhancements in the upper right hand box. Definition 1 captures these attributes and scores.

Definition 1 [Local Static Risk Score (LSRS) and Risk Categories]: The Local Static Risk Score is the sum of the following attributes and their assignable integer values;

Account Risk Range:[-5,+10], *Location Risk* Range: [-1,+10]

Business Risk Range: [-15,+20], *Product Risk* Range: [0,+5]

Here *Account Risk* is the sum of *Customer Risk* [-5, +10] and *Tax ID Risk* [+5]. The *Location Risk* is the Sum of *Primary Location Risk* [+2]. The sum of the Risks of Non Primary Locations, where each *Non Primary Risk* is a value in the Range [-1,+10]. The *Business Risk* is defined as the sum of *Business Primary Risk* [-3, +20] and *Business Nature Risk* [-15, +20]. The *Product Risk* is the sum of *Debit Activity Risk* [0, +5] and *Credit Activity Risk*: [0, +5].

The BSA risk score is the sum of the component risk scores of Account Risk, Location Risk, Business Risk and Product Risk. Each of these components risks are also sums of further sub components as specified in Definition 1. Possible computed value for a customer is an integer value for between -23 and +20. The details of risks used in Definition1 are as follows. The Account Risk is the Risk due to customer's reputation and a risk assigned due to providing / not providing a TAX ID. The Locations Risk is the sum of having multiple business Locations, and the risk associated with the Primary business Location. The Business risk is the sum of the risk due to the Principal Owner and the risk associated with the Nature of the Business. The financial product risk is associated with the debit and credit activities. We amended the factors of "product risk" in the BSA model to include a risk factor of the derivative market activity. We also reduced the risk weight of three factors in the "business risk" of BSA model from the original value of "+30" to the new value of "+20", as the total risk score of "30" is the cut-off for an alert to the management of the financial institution. The reduction of the weight of the three factors to "+20" is necessary to lower the aggressiveness of the risk model. Definition 2 categorize these risks as Low, Medium, High and Extreme and are again an extension of the values in [9].

Definition 2 [Categorizing Local Static Risk Scores]: Local Static Risk Scores (LSRS) are categorized as low, medium, high and extremely high based on range of the totally calculated score: ***Low*** [-23, 4], ***Moderate*** [+5, +14], ***High*** [+15, +30], ***Extreme Risk*** [+31, +153].

B. Accumative Static Risk Score

To compute the risk of transacting customers, in addition to Static Local Risk Score (LSRS), risk of recent transactions need to be taken into account. We propose a simplified mechanism of exchanging aggregate risk scores assigned to customer transactions, because a running average may not expose all the data of all transactions and therefore may not violate privacy. Formally, let *TRN(O,R)*, be a transaction with originator *O* and recipient *R*, and let $TRN_A \equiv <TRN_{A1}(x_1,y_1), TRN_{A2}(x_2,y_2),, \ldots\ldots, TRN_{An}(x_n,y_n)>$, listed in newest to oldest transaction order represent the last **n** transactions of A. Let $Partner_i(A,TRN_{Ai}(x_i,y_i))$ represent the entity other than A and $<LSRS(Partner_1(A,TRN_{Ai}(x_i,y_i)))>$be the LSRS values of partners of A in the last n transactions. Then recursively define the Exponential Moving Average (EMA) risk as: EMA(i) = $LSRS(Partner_i(A,TRN_{Ai}(x_i,y_i)))$* k + EMA(i-1) * (1 – k) where k = 2/(n+1) .

Definition 3 [Receiver's/Originator's Average Risk and Variance]: These averages are calculated by the bank that holds the account of entity A, it is done by calculating the exponential moving average of the LSRS of the last n transacting partners of A,

Let $\mathbf{EMA_i}$ *be* $LSRS(Partner_1(A,TRN_{A1}(xi,yi)))$* k + EMA(i-1) * (1 – k) where k = 2/(n+1), and where $A=x_i$ for all i<n.*

Let $\boldsymbol{Var_A}$ *be* $LSRS_A$– Average($LSRS(Partner_1(A,TRN_{A1}(x_1,y_1)))$, …, $LSRS(Partner_n(A,TRN_{Ain}(x_n,y_n)))$.

As with the Average, VAR_{Ai} computes the receiver's and originators risk based on the value of $Partner_i(A,TRANA_i(x_i,y_i))$. When $Partner_i(A,TRANA_i(x_i,y_i))$ = A= Receiver for all i<n. Then EMAi computes the receiver's risk and When $Partner_i(A,TRANA_i(x_i,y_i))$ = A= Originator for all i<n. Then EMAi computes the originator's risk. The *RA/OA* parameters assess the risks associated with the participation/involvement of an entity in the ML, by analyzing the affinity/role in the money-flow of a laundering process. The *RA* and *OA* used to calculate/keep a record of the historical activity and the divergence in pattern of receiving or sending funds. The pattern are used as an indicator for assessing a risk penalty, comparing the current *RA/OA* value with the value of 90 days and 180 days ago (*RA/OA*) indicate the transactional tendency of the entity.

We assign a penalty and reward system for entities so that an entity that continuously transacts in high and increasingly risky pattern is subject to penalties for having an increased LSRS, and vice versa. Thus our penalty and rewards system self-adjusts and the leverage provided by this self-adjustment avoid maintaining the risk value of an entity at a static level. The penalty can be set upon the needs of the financial institution and the regulations of the country, although optimal levels are shown in the formula below. The optimum penalty/reward are produced to allow entities to retain their old static risk levels in between one and **n** transactions. The criteria defined below indicate that if the aggregate risk sores is higher than 90 days ago which is higher than the same value 180 days ago this entity's transacting risk is on the increase.

Definition 4 [Penalties and Rewards]: We define $RA_{0M,}$ RA_{3M}, $RA_{6M,}$ OA_{0M}, OA_{3M}, and OA_{6M} to be respectively the current, three months, and six months old values of RA and OA value, for any entity. Let *RA-Inc*, *RA-Dec*, *OA-Inc* and *OA-Dec* be defined as ($RA_{6M}<RA_{3M}<RA_{0M}$), ($RA_{6M}>RA_{3M}>RA_{0M}$), ($OA_{6M}<OA_{3M}<OA_{0M}$), and ($OA_{6M}>OA_{3M}>OA_{0M}$) and Static Risk Penalty and Reward (SRPR) as:

(RA-Inc)/\(OA-Inc)/\(RA≥LSRS)/\(LSRS>35)/\(RV≥5)/\(OV≥5)=> SRPR=+5
(RA-Inc)/\(OA-Dec)/\(RA≥LSRS)/\(LSRS>35)/\(RV≥5)/\(OV≥5)=> SRPR=+3
(RA-Dec)/\(OA-Inc)/\(RA≥LSRS)/\(LSRS>35)/\(RV≥5)/\(OV≥5)=> SRPR=+3
(RA-Dec)/\(OA-Dec)/\(RA≥LSRS)/\(LSRS>35)/\(RV≥5)/\(OV≥5)=> SRPR=+2
(RA-Dec)/\(OA-Dec)/\(RA≥LSRS)/\(LSRS>35)/\(RV<5)/\(OV<5)=> SRPR=-2
(RA-Dec)/\(OA-Dec)/\(RA≤LSRS)/\(LSRS>35)/\(RV≥5)/\(OV≥5)=> SRPR=-3
((RA-Any)/\/(OA-Any))/\ (RA<LSRS) =>SRPR=0

A detailed rationale for this definition is described in [14]. The LSRS will be calculated every time the transaction occurs. For example, the first line of Definition 4 says that if conditions (1) "RA6<RA3<RA", (2) OA6<OA3<OA, (3) RA≥LSRS and (4) RV>OV>0 are met, the SRP of "+5" will be imposed. The RA is the primary factor that LSRS depends upon on to determine the penalty value due to the fact that receiving the funds is where the money laundering fraud starts. The penalty and reward point system will have the upper and lower bounds, in order to maintain the LSRS within its boundaries so that their accumulation will have a fix point (risk saturation point) in its decreasing or increasing trend. There is no need to apply the penalty on an entity that is in maximum risk levels of LSRS, as the purpose is to provide the transacting entity with the ability to reduce the risk.

Definition 5 [Accumulative Static Risk Score (ASRS)]: of an entity is the sum of the local static risk score and static risk penalty and reward. Thus, ASRS =LSRS + SRPR.

C. Accumulative Dynamic Risk Score

The dynamics of none-static risk scoring was designed considering the following criteria: (1) *Continuous scoring*: The score is calculated per every transaction. (2) *Automatic scoring*: Risk computation does not require the involvement of an expert. (3) *Correlation of past transactions*: Risk score correlate transactions with current one.

We have developed an algorithm to assigning weights to relations, the so-called Dynamic Relation Extract Algorithm (DREA) [14] that searches the SNA DB "SuspectWeight" for the detected past **n** ML activities of the entity A. The algorithm is similar to the algorithm used in the "SNA" module of MLEDF as explained in Section 2. Using this algorithm, we assign a risk weight for entity A, for each detected ML activity in the SNA DB "SuspectWeight" by adding parameters for each of the corresponding events resulting in the accumulative risk weight as follows: (1) For each detected MLS, add 5 to start/end boss couple, 2 for each boss/associate combination, and 1 for each associate/associate non-repeating combination. For each Missing-Trail add 3 to each associate non-repeating combination. The Full-Trails adds 3 to each associate combination and 10 to the start/end boss. Again these are our sample values that can be changed by any institutions. We omit an algorithmic presentation due to lack of space.

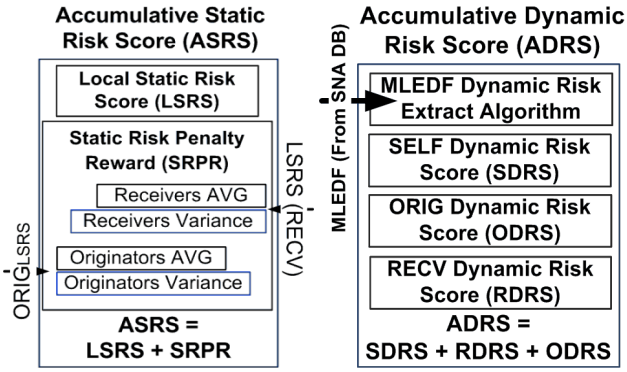


Figure 4. The Two Componenets of ML Dynamic Risk Model

We also compute a risk score named the Self Adjusting Dynamic Risk Score (SDRS) that assigns a risk weight to the transactional history of transacting entity (say) A in the database "Suspect-Weight" in the DREA algorithm that we refer to as DREA(Entity A)].

Definition 6 [Receivers' / Originators' Dynamic Risk Score (RDRS/ODRS)]: Calculates the aggregate risk weight, based on the relations history of the last **n** entities ($R_{1,...,}R_n$) funds receiving from, and the last n entities ($O_{1,...,}O_n$) funds originating to the entity A. The average weight of receiving /originating entities is obtained by calculating the average of DREA(R_1), .. ,DREA(R_n) and DREA(O_1), .. ,DREA(O_n). We produce ODRS and RDRS.

Definition 7 [Accumulative Dynamic Risk Score (ADRS)]: Of an entity is the sum of SDRS, RDRS and ODRS. That is ADRS = SDR+ RDR+ODR.

D. Accumlative Transaction Scoring Based on Dyanmic Risk

Static and dynamic risks are correlated to the analytics of transaction scoring, in order to identify transactions with high-risk score pertaining to ML, and to prevent transaction sequences from being executed. The correlations used in the dynamic risk scoring can be used to detect and track transactions belong of ML schemes.

Definition 8 [Accumulative Transaction Score (ATS)]:

The ATS is calculated as the average risk of (ADRS, ASRS, LSRS) of the two transacting entities.
Receiver-ATS=∑Receiver(ADRS,ASRS),
Originator-ATS = ∑ Originator (ADRS, ASRS).
ATS = AVG (Receiver-ATS , Originator-ATS).

Definition 9: Comprehensive Transaction Data (CTD): The triple (LSRS, ASRS, ADRS) is said to be the comprehensive transaction data (CTD).

Thus, our comprehensive dynamic risk model consists of two parts, computing the static risk, as by amending the BSA risk model [9] and computing the dynamic risk per every transaction. Figure 4 summarizes the two aspects of our risk model and the data used compute the individual components. As the figure shows, the static risk score is summarized in the accumulative static risk score (an enhancement of the BSA model) and the Dynamic risk score that takes the originators and recipients running averages of static risk scores and other properties of the transactions and SNA information to compute a dynamic risk value per each transaction. As stated, this value is fed back to the running averages and variances of this static risk scores. This latter step requires the financial institutions to share such risk estimates along with transactions.

V. VALIDATION

We used sanitized real-life cases to test and validate the dynamic risk model with MLEDF and transaction scoring. Our case studies are based on data provided from an organization we refer as Trusted Third Party (TTP), which is authorized to collect information and track records of financial exchanges.

A. Experimental Evaluation and Valiation of MLEDF

We introduced a three phase testing prototype to examine MLEDF and detection algorithms. All three phases focused on testing and validating the components of MLS, Full-Trails, and Suspicious-Trails. The first phase focused on testing all components and the other tests focus on Full-Trail and Suspicious-Trail components.

Test without noise: This test is designed to test every module of MLEDF, including detection algorithms and trail analysis modules. These tests evaluate the false positive rates (FPR) and false negative rates (FNR) by comparing the results of the test with the data feed that contains the patterns of single MLS, pair of MLSs, and Full-Trails. The desired result was to have a list of the validation result identical to the list in the data feed. We tested the efficiency to keep up with the speed of the data feed by using the time window feature in the StreamBase [10]. By setting the time window to glide over only one event at a time tick in the StreamBase system, we made the detection algorithms to run at the normal speed of one event at one time tick. By design, an algorithm that cannot attain the speed of event production will not be able to capture MLS events or the Full-Trail, thereby generating false negatives.

Each of the six detection algorithms were tested with their own data feeds in order to verify that we were able to detect a single event MLS without false positives and false negatives. The algorithm-specific dataset feed was generated using the built in feed generator working with our pattern specific event generator. Afterwards, we tested the "Missing-Trail" by feeding linked pairs of MLSs into the MLEDF. The linked/related pairs are randomly selected from the set of six types of MLS. As mentioned, any pair of linked MLS will make it to "Missing-Trail" and not into "Full-Trail", due to the required depth. Finally, we tested the detection and evolution of "Full-Trail"s by feeding trails generated from various laundering strategies used in our sample real-life cases.

The process of creating the "Full-Trail" started with creating an MLS type out of the six MLS types of Smurfing, Trading, DirtyEFT, Stock, Derivative, E-Money. Once the selection of first MLS is made, we create ta series of linked MLS based on conditions such as geography, amount of money, time, complexity of the schema and difficulty of tracking. The trails were created using different criteria and randomizing them using a normal distribution. We created the Full-Trail feeds using the generator to not exceed 10 levels of depth of linked MLSs. These trails were either a variant or a subsection of one of the real-life cases that were similar in terms of complexity and participants.

At the normal speed of one event at one time tick of the CEP system, the test result in zero false positive rates and false negative rates. It is highly improbable to get a false positive trail due to the business rules that define them, and due to the accuracy and granular level of linking transactions. We did not get any false positive rate (FPR) or false negative rate (FNR) in the MLS tests due to the synthetic nature of the data. When we increased the speed of the data generated to 10 times and 100 times the normal speed, we observed a FPR and FNR in the objects detected in the Full-Trail algorithms. Increasing the speed of processing did not produce FPR and NFR for a single MLS, but it produced FPR and FNR for MLS pairs at speeds that were multiples of 100s. The term "object" in this graph refers to the three different patterns of single MLS, pair MLS, and Full-Trail in the proprietary test of the specific object (Object in the first pattern tests to the first pattern single MLS, in the second to MLS pair, in the third to Full-Trail). The values of FRP and FNR reflect the number of falsely detected objects.

Test with subtle noise: This is the most relevant accuracy test of our detection algorithms. The goal of this test was to mislead the detection algorithms by generating false positives and false negatives synthetic data. The test had three separate phases: injecting the scheme participants, injecting subtle transactions, and inserting similar MLSs. A subtle transaction is a transaction with ±5% of an actual transaction amount in a MLS. A similar MLS is identical to a real-life MLS with the same set of participants but with the MLS value is ±10% of the laundered amount of the actual MLS. The injection speed was set to normal processing speed, 10 times faster, and 100 times faster. The test of injecting transactions and MLSs is setup considering each MLS type. For example, in the test of smurfing, we created only smurfing MLS and smurfing transactions that can extend vertically up to 20 levels of depth and horizontally to 30 levels of depth. When we were generating the MLSs our measures did vary based upon the MLS. We did not use artificially created none-real life cases. For example, we did not use a smurfing MLS with 100 levels deep, as that is uncommon and impractical to launder money. We also did not inject other types of MLSs into the injection test of a specific MLS. However, in the Full-Trail test, we injected all types of MLS because by design, a Full-Trail is required to have different types of MLS under the same Full-Trail.

The test produced low FNR and low FPR for transaction and MLS injection when the phases were executed at normal processing speed. Those rates increased in the phases when tests were executed at faster processing speed. One way to imitate the data rate of real production environment is to run the CEP tests at a faster rate, thereby overloading the system with processing and analytics while attempting to keep pace with the data stream. The goal was to evaluate the effectiveness of "Full-Trail" detection when the system absorbs data at a higher rate while performing the analysis. Due to the design methodology of detection algorithms and the complexity of the business rules of MLS detection, their false detection rates stayed at low levels (less than 5%) even with injection similar transactions and MLSs, at a higher data-feed speed (1000% and 10000% speed).

Meeting the design principles, the "Full-Trail" and "Suspicious-Trail" results remained at low rates for both false positive and false negative. Therefore, all the subtle single MLS created with injected data ended in the "Missing-Trail", where they did not exceed the depth of 3 consecutive MLSs. Some reasons for this success in trail analysis and avoiding any negative impact are (1) MLEDF is designed in a strict and granular method, especially for matching MLSs within trails, (2) SNA is used in the trail analysis algorithms, (3) Adopted the criterion to follow the direction of the flow of the laundered-money. MLS is not expected to terminate with funds remaining in the account. The money must flow in some direction in order to be laundered, or must be withdrawn by the launderer. The Figures 5, 6 and 7 show the results of the number (quantity) of the transactions resulted in false positive and false negative, as explained in the previous paragraph. The figures show the number of FP and FN patterns of each phase from the three injection phases of the Test II, along with the results from running the test at different speed (1000% and 10000% speed).

Test with longer synthetic full-trails: This was the hardest level of performance testing of the system and accuracy-testing of the detection algorithms we carried out. In this test, the dataset was permutated over a repository of different real-life cases. Afterwards, the dataset was combined with randomized MLS to generate deep vertical levels of "Full-Trail"s and "Suspicious-Trail"s. The randomization followed the same principles used in Test II's injection testing. The test was designed to assess the performance of MLEDF in capturing real-life data and analyzing them on the fly. The desired test result was to generate low FPR and FNR. The test module generated all synthetic data from real-life cases and tests were similar to real-life scenarios, considering that there are limited ways to manipulate a MLS. The test program functions as follows: (1) Set a trail depth. The program enters a loop and builds a trail by choosing a first scheme from of each MLS type at random, as it was described in Test I in building the Full-Trails. (2) The loop continues by creating an MLS that can be linked by funds, time, location and complexity to the current MLS. We repeated the step above with the exception of not creating any Smurfing MLS for the rest of the levels. (3) The permutation continues until the system reached the last level, where we always choose an MLS of type DirtyEFT with a withdrawal in order to generate the trail termination point, as by definition a trail will end with the withdrawal of money. (4) The test were repeated the process of trail generation forever at the maximum possible speed. (5) The testing module saved the arrival time of the last DirtyEFT and subtracted that from the build times of the trail, thereby obtaining Milliseconds difference in trail processing times.

Our data was generated for worst-case scenarios to ensure that they are more complex and the performance was evaluated only in most resource consuming cases. Displayed results represent the performance of data generated without any repetitive bosses or associates. Hence, the dataset consumes a significant number of resources.

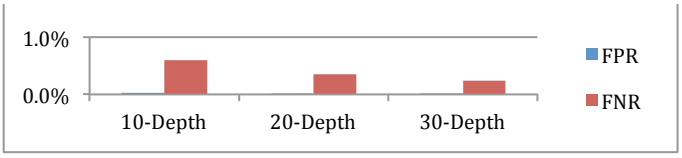
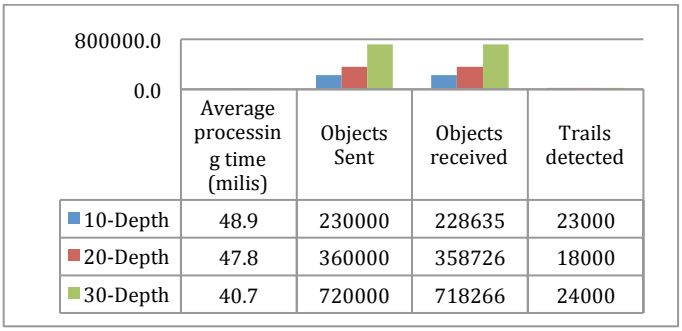


Figure 5. False Positive and False Negative Percentages of Test III



	Average processing time (milis)	Objects Sent	Objects received	Trails detected
10-Depth	48.9	230000	228635	23000
20-Depth	47.8	360000	358726	18000
30-Depth	40.7	720000	718266	24000

Figure 6. Number of Detected Trails in Test III for Faster Data Rates

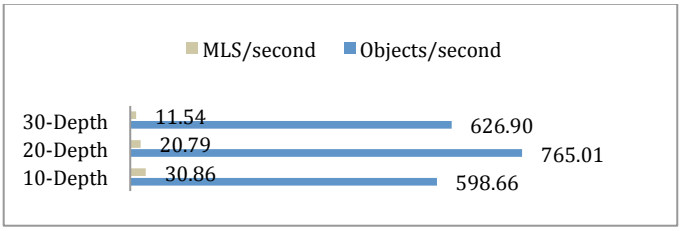


Figure 7. Pattern Generation Speed for Test III

B. Experimental Evaluation and Validation of Money Laundering Dynamic Risk Model

Test Methodology: We introduced a four phase risk model testing prototype to examine the three different versions of static risk model, and the dynamic risk model: (1) T1: Using standard average instead of the exponential average in calculating static risk. (2) T2: Using exponential average, but applying only penalty and no reward in calculating static risk. (3) T3: Using exponential average, apply both penalty and reward in calculating static risk. (4) T4: Using detected schemes, from the output of MLEDF, to produce dynamic risk scores.

Injected Data Phases: Four different types of transactions were injected in each of test phases with LSRS value of (10, 20, 30, 40) in the transactions of each test.

Test Goals: (1) Produce risk levels above certain threshold for continuously riskily transacting entities. (2) Effectiveness when certain patterns (all high risk or all low risk) were injected, (A) Does the ADRS/ASRS saturates at some fixed-

point level? (B) False Positive (FP): ADRS/ASRS continue to grow towards the high risk level of the continuously injected data (C) False Negative (FN): ADRS/ASRS deviates towards the low risk level of the continuously injected data (D) Maintain a desired risk level for bad entities even if they deliberately transact with good entities, in order to lower their risk profile.

Results: The dynamic risk model (T4) produces FP for transactions of none-MLEDF entities. The rate was less than 5% and that was satisfactory considering the large amount of transactions. This is advantageous compared with risk models that do not assess the risk of being involved in MLS, considering the factors of increasing risk scores of MLEDF entities.

Validation Statement: We used the StreamBase Studio [10] platform in each test of (T1, T2, T3, T4) and with each of the four data injection phases (by only injecting entities did not exist in MLEDF). The false negative rate was below 1% in phase 1 of all tests, and 0% in remaining three phases of data injection for all tests. The false positive rate was below 5% for T4, and lesser for other the three static tests (T1, T2, T3). In test T4 and with each of the four data injection phases by injecting entities did exist in MLEDF. The false negative rate for T4 (When only injecting entities that are already detected by MLEDF) is the highest at 11% when entities with high static risk (of LSRS 30) are injected in phase 4, then at 9% in phase 1 when high static risk score (of LSRS 25) are injected, then at 8% in phase 3, and finally at 3% in phase2 when low risk score (of LSRS 10) is injected. The false negative rate is 0% in all phases of test T4. Table V summarizes our findings. Figure 9 shows the false positive and false negative rates for injecting MLS's with 10, 20, 25 and 30 LSRS values.

TABLE V. NUMBER OF TRANSACTIONS WITH FN AND FP RISK

Transaction Injection/Test Type	T1	T2	T3	T4
Total Generated Transactions	240387	240387	240387	240387
Originators not from MLEDF	227	227	227	227
Originators from MLEDF	59	59	59	59
Unique Receivers	936	936	936	936
Injected MLEDF Transactions	9851	9851	9851	9851
FP- Phase1- Growing Risk	9	17	14	26
FN- Phase1- Declining Risk	0	0	2	0
FP- Phase2- Growing Risk	2	8	3	12
FN- Phase2- Declining Risk	6	1	2	0
FP- Phase3- Growing Risk	7	14	10	21
FN- Phase3- Declining Risk	0	0	0	0
FP- Phase4- Growing Risk	14	28	20	44
FN -Phase4- Declining Risk	0	0	0	0

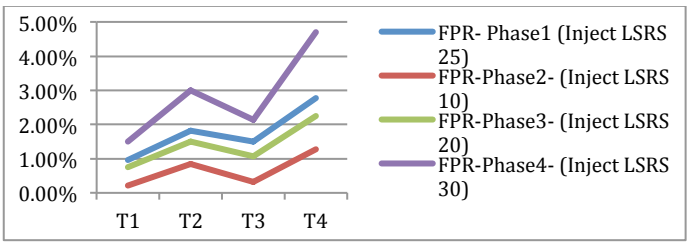


Figure 8. False Positive Rate for each risk models after data injection (none-MLEDF Entities)

VI. CONCLUSIONS

We implemented a multiphase, multilevel, and multi-component methodology to detect evolving money-laundering schemes using known methods, influenced by economic factors. We have created a framework to detect the evolution of MLS and implemented a system to include SNA for detecting and linking related ML networks. This linkage will function properly even when all evidence is unavailable. We defined the choreographies that could be used to detect the evolution of the sophisticated MLS. We have shown how to detect and capture the evolving and complex trails of MLS using SB.

We enhanced the BSA inspired static risk with aggregated static risk, to reflect the changing dynamics of the ML and its consequences on the risk calculation. Our risk model factors in the initial account-opening risk as well as subsequent transactional risks, and it presents a risk score that is valid within and outside the boundaries of a single financial institution. We extended the static risk model to develop a MLEDF-dependent risk modeling, in order to produce a comprehensive ML risk modeling in combination with the aggregated static risk model. The aggregated static risk will be completed with integration of the MLEDF-dependent risk modeling, which captures the hidden, and dynamic, relations among none-transacted entities. Such a risk model is used to create a valid and accurate transaction scoring system to be used in a ML prevention system.

REFERENCES

[1] C. Weinstein, W. Campbell, B. Delaney, G. O'Leary, "Modeling and detection techniques for Counter-Terror Social Network Analysis and Intent Recognition," Aerospace conference, IEEE, 2009.

[2] Financial Action Task Force, "Global Money Laundering & Terrorist Financing Threat Assessment Annual Report," February 2013.

[3] D. Schwartz and T. Rouselle, "Using social network analysis to target criminal networks," Trends in Organized Crime, 2008.

[4] Financial Action Task Force, "Money Laundering & Terrorist Financing Risk Assessment Strategies," June 2008.

[5] Wolfsberg Group, "Guidance on a Risk Based Approach - Wolfsberg Principles", 2006.

[6] Scor Inc, "The risk of money laundering: Prevention, challenges, outlook", 2008.

[7] Kount Inc, "Dynamic Scoring and Rescoring", 2011.

[8] M. Mehmet and D. Wijesekera, "Detecting the Evolution of Money Laundering Schemes," IFIP WG 11.9 Conf. on Digital Forensics, 2013.

[9] BankersOnline, "Risk Rating - Commercial Risk Rating Spread-sheet", http://www.bankersonline.com/tools/bc_commercialriskrating.xls

[10] StreamBase, 'Powerful Real-Time Architecture for Today's High Performance Modern Intelligence Systems', Federal Government, Defense, and Intelligence Applications, 2012.

[11] FinCEN (2013), *Answers to Frequently Asked Bank Secrecy Act Questions*, accessible via http://www.fincen.gov/statutes_regs/guidance/html/reg_faqs.html .

[12] FinCEN (2013), *Bank Secrecy Act Requirements - A Quick Reference Guide for MSB,* http://www.fincen.gov/financial_institutions/msb/materials/en/bank_ence.html .

[13] Office of Foreign Assets Control (OFAC) (2013), *Designated Nationals List (SDN)*, http://www.treasury.gov/ofac/downloads/ctrylst.txt and http://www.treasury.gov/ofac/downloads/t11sdn.pdf .

[14] M. Mehmet, Money Laundering volution Detection, Prevention and Transaction Scoring, PhD Dissertation, George Mason University, 2013.

[15] Hawala, http://en.wikipedia.org/wiki/Hawala, refereed on 10/13/13.

Extraction of Semantic Activities from Twitter Data

Aleksey Panasyuk
Air Force Research Lab
Rome, NY, 13441
Aleksey.panasyuk@us.af.mil

Erik Blasch
Air Force Research Lab
Rome, NY, 13441
Erik.blasch.1@us.af.mil

Sue E. Kase
Army Research Lab
APG, MD
Sue.e.kase.civ@mail.mil

Liz Bowman
Army Research Lab
APG, MD
Elizabeth.k.bowman.civ@mail.mil

Abstract— With the growing popularity of Twitter, numerous issues surround the usefulness of the technology for intelligence, defense, and security. For security, Twitter provides a real-time opportunity to determine unrest and discontent. For defense, twitter can be a source of open-source intelligence (INT) information related to areas of contested environments. However, the semantic content, location of tweets, and richness of the information requires big data analysis for understanding the use of the information for intelligence. In this paper, we describe some results in using twitter data to determine events, the semantic implications of the results from the data, as well as discuss pragmatic uses of twitter data for multi-INT data fusion. The results collected during the period of Egypt Arab spring conclude that (1) many tweets are clutter or noise in analysis, (2) location information does not always convey the accuracy of the information, and (3) the aggregate processing of the twitter data results in real-time trends of possible events that warrant more conventional information gathering.

I. INTRODUCTION

Over the last decade, there has been a surge of the use of constrained messages sizes in 140 characters or less, known as "tweets." The popularity of tweets has three emerging issues: (1) big data as the number of users grows, (2) semantic extraction of meaningful content from cryptic phrases and non-standard terminology, and (3) the large amount of semantic clutter that reduces the signal-to-noise ratio of identifying salient content (e.g., key words of phrases).

A. Twitter as a source of Intelligence

While the use of open source information becomes popular such as Facebook, imagery, and text; it is well established that tweets are being used by anyone anywhere from distributed mobile platforms. The presentation of different semantic formats and the number of users require pragmatic approaches to searching and deriving meaningful content from tweets. Meaningful content is further exacerbated as the source of tweets does not always correspond to the location of the user; however, timing and general trends over many users can determine the status of an emerging event.

Twitter data, while popular, suffers from various content issues that have to be solved with advanced and tailored methods. Examples of problems include users with hidden meanings, masked source of origin, possible deceit and deception, as well as non-descriptive and non-important discussions of social issues (e.g., where to go for dinner). However, tweets do provide a forum where users can express their social and political views, news reports of immediate actions that are not available to the regular media, and links of semantic content such as to video collected and posted to the web from cell phones.

It is the interplay between the availability and enormity of tweets to that of extraction of meaningful content that is derived from users close to the action. Tweets provide reports that are not available from other intelligence sources of information.

B. Twitter Semantic Extraction

An approach taken by most search engines over twitter data is to organize documents and their terms in a Vector Space Model (VSM) [1]. A Vector Space Model is a two dimensional array. The rows of this array are a list of terms from all documents that a user is searching through. The columns are the names of documents. The VSM ranks all terms using frequency analysis utilizing the bag of words hypothesis. Bag of words hypothesis states that two documents tend to be similar if they have an equivalent distribution of analogous words [2]. In this way, a search engine query can be seen as a vector of terms which can be used with a VSM in order to find documents that are closest to this vector via some distance measure.

With successful implementation of VSM by the search engines, researchers have attempted to apply VSMs to other areas of natural language processing (NLP). For a long time it had been considered that to understand the meaning of words it is enough to consider statistical word usage, the so called statistical sematic hypothesis [3,4]. The benefit of VSMs is that they easily consume large amounts of data and require far less labor than other approaches [5]. For example, Rapp [6] developed a vector representation of word meanings mainly from British National Corpus. The British National Corpus is not a lexicon but is simply a text corpus containing 100 million words annotated with parts of speech (www.natcorp.ox.ac.uk/). Rapp's VSM was used on multiple choice questions from Test of English as a Foreign Language (TOEFL) achieving a score of 92.5% where the average human score was 64.5% [6]. TOEFL is a well-structured text making preprocessing and identification of terms an easy task.

However, Twitter is more complicated as the text is unstructured.

Twitter has a lot of informal and abbreviated text. For example, current tools, while practical on news articles and similar types of well written documents, perform quite poorly for Part-of-Speech (POS) tagging and Named Entity Recognition (NER) when applied to tweets. The accuracy of tools falls from 97% accuracy for news articles to about 80% for tweets [7]. In [8], Finin *et al*. experiment with crowdsourcing for POS tagging on tweets. Crowdsourcing is made available by a service such as Amazon's Mechanical Turk which allows for tasking and collecting results from a "crowd" of people that are willing to do the work by hand. Others [9, 10] propose lexical normalization of tweets which may be useful as a preprocessing step for the upstream tasks like POS tagging. In [11], Gouws *at al*. try to properly tag parts of Twitter speech. They mention that it is hard to tag words within Twitter because of the conversational nature of text, lack of conventional orthography, and limit of 140 characters. Messages on Twitter are filled with grammatical errors, abbreviations, slang, words in another language, and URLs. The authors take a number of steps to give them an advantage such as using the Metaphone algorithm [12] in order to remove alternative spelling for words, developed expression-style rules for capturing known structures like URLs, keeping capitalized words that follow an expected distribution, as well as using known lexicons such as WordNet in conjunction with their algorithms, etc. In trying to understand statistical semantics in Twitter, the authors use unsupervised word representations as extra word features. 1.9 million features from 134,000 unlabeled tweets are used to construct these distributional features via an approach outlined in [13]. Even though the training set is limited to 1000 records (tweets), the unsupervised word representations capture enough content to achieve nearly 90% accuracy on the 500 testing records.

To perform further analysis such as sentiment analysis, it would make sense to perform all of the steps in the papers cited i.e. set of tweets needs to be found, the features from tweets are extracted, preprocessed, tagged, and statistical analysis is performed. Sentiment analysis can be used in order to identify anger, tension, and other emotions that may be tied to significant offline events [14, 15]. Tweets can even be used for predicting events like earthquakes [16] and box office sales [17]. Twitter has a lot of data, about 400 million tweets per day, which is beyond what human beings can handle even with crowdsourcing. Finding relevant text is becoming increasingly challenging such that there is a growing need for automatic text understanding that scales to the Web. There are systems known such as open Information Extraction (IE) systems that are being developed to address text understanding [18, 19, 20], but in order to use such systems we need to know the features of interest.

The rest of the paper is as follows. In Section II, we describe the use case of Twitter data from the Egyptian uprising. In Section III, we discuss how the data is structured. Section IV and VI describe analytics and visualization, respectively. Finally, Section VII describes conclusions.

II. TWEETS FROM THE ARAB SPRING SCENARIO

The Egyptian uprising of 2011 is an example of an important historical event which has been captured via social media sites such as Facebook and Twitter. Some argue that without social media, like Twitter, the uprising would have not achieved the same level of success [21, 22]. Social media allowed countless participants to be involved. Twitter has become an important social media site since its inception in 2006. It is a micro blogging service which allows users to post messages up to 140 characters in length. Once a message is posted, any twitter user in the world can see it, repost it, and reply to it. A user may search for messages based on topic or person of interest. A user may choose to "follow" another user which will cause all of the messages posted by a user that is being followed to be displayed on that user's Twitter timeline.

In regards to Egypt, January 25th 2011 had become known as the "day of rage" with protests in Cairo. Social media and Internet played such a key role that the Egyptian government had begun limiting Internet access on January 27th [23]. Egypt related topics continued circulation until February 11 when President Mubarak resigned. During the course of events, it was noted that information was coming from tweets, but the intelligence sources were not mobilized to use the technology and even if available, to what extent that content could be gleamed from the experience of multiple users presenting tweets.

In this paper we attempt to analyze 738,717 tweets from that time period in 2011. The Egypt Twitter data has been grouped into 10 classes by Army Research Lab (ARL) [24]. The 10 classes come from ranking the Twitter data using the Tri-HITS model described in [25]. Tri-HITS paper describes an algorithm for ranking tweets not only based on the textual content within the tweet but by also considering the referenced web documents and popularity of users. The results of the Tri-HITS model show improvement over popular algorithms such at TextRank [26]. Once the Egypt tweets are ranked they are equally broken down into 10 groups with first group being the tweets that had a ranking that beat 90% of other tweets, second group being the tweets that beat 80% of tweets but are not part of first group, and so on. Given these initial classes we investigated whether the features behind those classes made sense and could be used for deeper searches by the analyst. Our interests for this data had been whether we could use it as a source for proactive situational analysis.

In relation to the above methods, our novel approach is to apply a VSM model to the groupings made by the Tri-HITS model in order to extract top one hundred features associated with each grouping. The top features that are extracted can be used for evaluating the quality of grouping and can be used by the analyst when searching for similar events. The algorithm is fast and straight forward to implement and does not require human in the loop involvement. To the best of our knowledge we are not aware of papers applying a VSM model on rankings generated with Tri-HTTS model for the Egypt data.

III. STRUCTURING DATA

The first task was to properly structure the data within a MySQL database. This is the environment that we have been using:

- Windows 7 64 bit
- Eclipse 4.2.0 with Python IDE plugin (PyDev)
- Python 2.7.x with MySQL-python connector
- MySQL 5.5 with MySQL Workbench 5.2 CE

Each tweet is limited to 140 characters and is associated with some class label. Figure 1 shows the basic tweet to class label structure. There are a total of 10 classes c_0, c_1, …, c_9.

```
1 • SELECT * FROM egyptuprising2.tablecontainingrawdata;
```

Filter: Export: Autosize: Fetch rows:

class	text
c9	@Panoptique Is Stephen Cohen nuts? Deaf? Blind? Stupid? Related to Sarah Palin?
c9	#Imagine what would happen if #censorship occurred in the United States? #foodforthought #Egypt #Jan25 #change
c9	Egypt.
c9	@djeratic Egypt?
c9	#Egypt Ah! Now I know why the military is in Sharam El Sheik! With Israeli agreement! Hosni Mubarak is there! Not Cairo!
c9	Proof I miss being in Seattle.. Watching the news on Cairo I keep thinking... What's wrong with channel 7! If only they still had JP Patches
c9	Helicopters are still roaming the air, protesters are still in Tahrir. I hear them chanting still
c9	so what to pack ....
c9	Morning all- gym- done bring on the day
c9	For those interested, this is the #Google search I use to get these messages from #Egypt http://bit.ly/g5l4pZ
c9	What will happen today? #Egypt
c9	'Good morning tweeps have a super day \u263a
c9	'La \u201cMarcha del mill\xf3n\u201d parte desde Tahrir y va hasta el Palacio Presidencial, inicia a las 9:00 hora de #Egipto 7:00 GMT
c9	#Egyptians #Cairo #Mubarak
c9	Another sweaty day Ethekwini. Love it, wouldn't have it any other way!

Figure 1. Tweet to Class Data

Figure 2 shows a vision for all of the main steps listed in the referenced papers. The first step is to *find information of interest* whether it is related to some event, an organization, a product, etc. Open IE systems can help retrieve the data we are interested in, if we have a broad enough set of terms that cover the topic of interest. The event of interest for us is the Egypt uprising data supplied by the ARL [24]. We don't use an IE system in this paper, but it would be the goal to use the terms extracted from this research with an IE system in the future in order to find related events.

The second step is to *tokenize*, i.e. extract features. Most often terms of interest are separated by white space, but researchers need to consider how they want to treat URLs, punctuation, and multiword features such as "daylight savings". We had two approaches. Our first approach had been to simply use white space as delimiter, join on punctuation, and disregard features over 50 characters in length (this gets rid of most websites). Our second approach was to focus on specific topics and people on Twitter:

Tokenize Approach A. Feature = anything that is no more than 50 characters in length and that contains only digits and ascii_letters i.e. any other characters are removed.

There were a total of 782,713 features using this approach.

For example:

http://www.google.com is converted to → httpwwwgooglecom which becomes our feature.

Tokenize Approach B. Feature = Twitter hashtags (Twitter topic that begins with "#") and Twitter at-mentions (at-mentions begin with "@").There were a total of 106,322 features using this approach.

For example:

#egypt and @youtube would be the structure of our features.

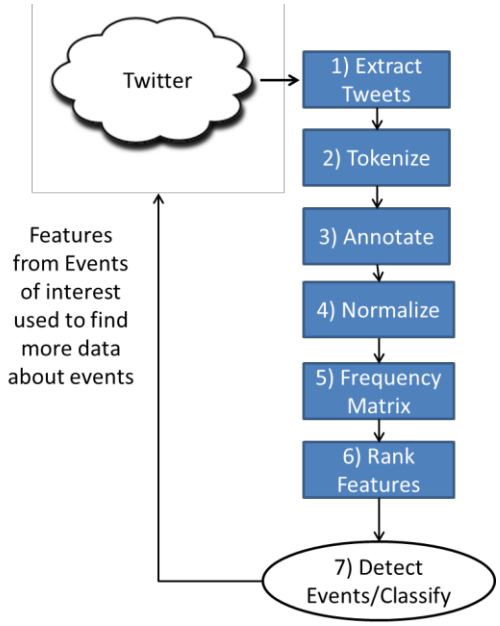


Figure 2. Main Steps for processing

The third step is to *normalize*. Normalizing reduces similar features. For instance it is common to use a stemmer in order to turn words like fixing, fixed, fixer → fix. The Metaphone algorithm mentioned in [11] will map words from a set like {thangs thanks thanksss thanx thinks thnx} to a single key, but sometimes this is not desirable as {war we're wear were where worry} are also mapped to a single key. A researcher may also choose to remove common stop words like "the". Normalizing will typically increase recall (when system identifies a relevant tweet as relevant), but decrease precision (when a tweet that is identified as relevant is truly relevant). In this paper, the only normalization we do is to turn everything to lowercase and consider only printable characters.

The fourth step is to perform *annotation*, for example roll can be tagged as a verb roll/VB (to rotate around an axis) or

a noun roll/NN (a small loaf of bread). Annotation is the inverse of normalization so it tends to improve precision and decrease recall. Annotation can be performed using well established lexicons that contain basic rules of grammar for such operations; examples include WSJ and Brown corpora as well as WordNet and Moby lexicons. In this paper we have not attempted any annotation.

The fifth step is to use the features in a *frequency matrix*. It begins by recording how many times each feature appears in each class. Table 1 is the result of an SQL query which orders features by total times used over all classes using tokenizing approach A.

feature	c0	c1	c2	c3	c4	c5	c6	c7	c8	c9	Total Times Used
egypt	35315	33624	39010	37230	32318	18043	16740	18092	18481	20441	269294
the	13506	14938	23251	22736	18969	12753	15956	17324	17224	17970	174627
in	19952	19225	23614	21073	17409	10044	12205	13437	13353	14429	164741
to	12106	11507	16759	17178	14557	10164	13285	14364	14575	14998	139493
tahrir	21220	19984	21232	13619	7300	2323	7745	8813	9148	10007	121391
cairo	21213	15749	17590	16029	10397	3181	5868	6736	6780	7648	111191
25-Jan	15708	14561	16954	13976	9218	5093	7094	7878	7808	8432	106722
of	9882	10240	15503	14815	11854	7027	8054	8704	8617	9126	103822
a	6527	6706	10892	12640	11204	8137	10694	11381	11517	11567	101265
…	…	…	…	…	…	…	…	…	…	…	…

Table 1: Feature Counts for each Class

	c0	c1	c2	c3	c4	c5	c6	c7	c8	c9
Classical Accuracy	0.19078	0.141639	0.158196	0.144157	0.093506	0.028608	0.052774	0.06058	0.060976	0.068783
Percent Contribution	0.191419	0.141558	0.157702	0.143607	0.093232	0.029631	0.053232	0.060323	0.060845	0.068451
TF-IDF	0.252498	0.1849	0.206659	0.187659	0.120601	0.037859	0.068309	0.07751	0.078192	0.088101

Table 2: Example Feature Scores

The sixth step is to *rank features*. The counts can then be used to calculate probabilities and to rank features. Classical frequency only considers how probable a term is to occur within a class. For example consider that the feature "to" was seen 1000 times for class 1 and 2000 times for class 2. Classic accuracy is going to be: "to" appears 1000/3000 = 33.33% of time in class 1, and 2000/3000 = 66.66% of time for class 2.

Classic accuracy could be taken to mean that the feature "to" is associated with class 2 with 66.66% accuracy. But consider that class 1 had a total of 5000 records and class 2 had a total of 10000 records. This additional information tells us that "to" had appeared in every record of class 1 and every record of class 2. Hence "to" is not a relevant feature.

Instead of using classical frequency, most papers use the term *frequency multiplied by inverse document frequency* (TF-IDF) [27]. In this way the greatest ranking is when the feature is frequent for a particular class and not frequent in all other classes, calculated as:

$$\text{TF} - \text{IDF for feature i in class j} = \left(\frac{\text{Total times feature i appeared for class j}}{\text{Total number of tweets in class j}}\right) * log\left(\frac{\text{Total number of classes}}{\text{Total number of classes that have feature i}}\right)$$

Besides using the TF-IDF measure we rank features by calculating how each feature contributes to each class using the following percent contribution formula:

$$\text{Feature i contribution to class j} = \frac{\left(\frac{\text{Total times feature i appeared for class j}}{\text{Total number of tweets in class j}}\right)}{\sum_{k=0}^{n-1}\left(\frac{\text{Total times feature appeared for class N}}{\text{Total number of tweets in class N}}\right)}$$

(where n is the number of classes involved)

Both measures consider the number of records in all classes which is better than using classical frequency. Using the percent contribution formula on the example from above, we see that class 1 and class 2 evenly contribute to this ratio, i.e.:

denominator = 1000/5000 + 2000/10000 = 2000/5000

%contribution by "to" feature from class 1 =
(1000/5000)/(denominator) = 50%
%contribution by "to" feature from class 2 =
(2000/10000)/(denominator) = 50%

TF-IDF will rank these two classes as equals as well:

(1000/5000)*(log(2/2)) = 0
(2000/10000)*(log(2/2)) = 0

Table 1 is used to generate measures for each feature using classical accuracy, percent contribution, and TF-IDF. Table 2 shows the calculation for the three measures for the feature "cairo". The examples had been shown using data from tokenization approach A, but the same approach and tables are produced when performing tokenization approach B.

IV. RANKING FEATURES

Given a score for each feature, we are able to go through all of the original tweets and classify the tweet using the feature within the tweet that has the highest score. We keep track of how many times a feature is used. Ordering on times that the feature had been used to predict a class gives us a ranking of all the features. Table 3 shows top features used by the three measures.

Classical Accuracy		Percent Contribution		TF-IDF	
feature	timesUsed	feature	timesUsed	feature	timesUsed
i	15508	i	15233	egypt	284295
square	5461	square	5492	in	87731
im	3547	me	3450	the	54929
me	3487	im	3446	cairo	48124
protesters	2872	protesters	2870	tahrir	39523
mubarak	2603	mubarak	2595	i	37401
revolution	2302	revolution	2352	to	19928
cairo	1603	cairo	1714	rt	14162
tahrir	1410	tahrir	1448	a	12600
egypts	1384	egypts	1395	el	9096
news	1367	news	1375	25-Jan	7062
protests	1359	protests	1346	of	6813
jazeera	1213	jazeera	1213	is	6716
president	1175	president	1180	and	6361
…	…	…	…	…	…

Table 3: Top Features used in the Classification of Tweets (tokenization approach A)

From the table, we see that TF-IDF has identified many stopwords as important because those features are a big percentage of the tweets. We see that percentage contribution used the feature "i" less than classical accuracy (score for feature "i" is slightly less by percent contribution) and the feature "square" more than classical accuracy (score for feature "square" is ranked slightly higher by percent contribution. Percent contribution should be more accurate because it takes into account number of records within each class when calculating its scores.

Table 4: Shows the same type of analysis performed on hashtags and at-mentions (tokenization approach B).

Classical Accuracy		Percent Contribution		TF-IDF	
feature	timesUse	feature	timesUse	feature	timesUsed
#egypt	48353	#jan25	33848	#egypt	165618
#tahrir	30509	#tahrir	32475	#jan25	29356
#jan25	11475	#egypt	27403	#tahrir	11876
#mubarak	11129	#mubarak	10771	#cairo	5682
#cairo	9385	#cairo	10213	@ghonim	3127
#25jan	3996	#25jan	4187	@addthis	2834
@youtube	3668	@youtube	3660	@youtube	2757
@addthis	3002	@addthis	2972	#news	2469
#news	2601	#news	2652	#tcot	1489
@ghonim	2261	@ghonim	2269	#mubarak	1370
#bahrain	1995	#bahrain	1971	#reasons mubarak islate	1341
#freeegypt	1795	#free egypt	1810	@fatmega loman	1317
#yemen	1700	#yemen	1656	@sand monkey	1266
#egipto	1380	#egipto	1374	#bahrain	1207

Table 4: Top Features used in the Classification of Tweets (tokenization approach B)

When looking at top 100 features associated with each class there is a clear difference between classes as we go from class c_0 to class c_9. Features seem to be going from clear topics during the Egypt revolution to features corresponding to personal tweets. Tables 5 and 6 show percent contribution measure top features for classes c_0, c_1, c_8 and c_9 for tokenization approach A and B (respectively).

c0_Feature	c1_Feature	…	c8_Feature	c9_Feature
protesters	square	…	me	i
cairo	jazeera	…	ive	ppl
tahrir	yemen	…	love	anyone
egypts	mubaraks	…	you	so
news	live	…	think	know
president	clashes	…	damn	morning
cairos	http English aljazeera netwatchnow	…	quoti	ok
military	al	…	haha	going
reuters	humidity	…	terradaki	go
25-Jan	algeria	…	whats	swear
hosni	cooper	…	am	omg
thousands	resignation	…	nretweet	khouly3
…	…	…	…	…

Table 5: Top Features Percent Contribution (tokenization Approach A)

c0_Feature	c1_Feature	…	c8_Feature	c9_Feature
#jan25	@youtube	…	@dima_khatib	#fb
#tahrir	#yemen	…	@elazul	@nevinezaki
#cairo	#libya	…	@alyaagad	@etharkamal

#25jan	@ajenglish	…	@mamoudinijad	@gsquare86
@addthis	@waelabbas	…	@khalawa69	@monasosh
#news	#algeria	…	@amrwaked	@nadaauf
#freeegypt	@salmaeldaly	…	@saraaayman	@khouly3
#aljazeera	@ajelive	…	@theonly warman	#icantdateyou
#feb11	#weather	…	#grammys	@travellerw
#alarabiya	@huffington post	…	@terradaki	@mosaa berizing
#ghonim	@shmpongo	…	@litfreak	#iheard
@guardian	@washington post	…	@marionnette90	#mbmemories
@addtoany	@adel_salib	…	@samiyusuf	#prayforegypt

Table 6: Top Features Percent Contribution (tokenization Approach B)

Having found ranked features for each class, the analyst can verify if the features captured make sense and use those features in order to filter and collect more data from Twitter.

V. Analyzing Overall Error

Twitter is a large noisy data source. There are 400 million tweets a day with most of the messages not relevant to the analyst. Simply grabbing a lot of data and trying to fit a model to the messages is not relevant. An analyst needs to first understand how to query Twitter just as an ordinary human being knows how to query the World Wide Web. Querying Twitter is equivalent to understanding the types of features (query terms) to use. We have illustrated a means of ranking features and then using those features for classifying a 10 class problem. Calculating accuracy is simple in the sense that we can just count how many times we have accurately identified a record vs. number of records attempted. The features are ranked by the overall accuracy for the 10 classes achieving 0.695% for percent contribution error, but the final classifier had hundreds of thousands of features that appeared only once. For this reason we choose to look at only the top 1000 features, with accuracies shown in Table 7 and 8.

Classic Accuracy	24.46%
Percent Contribution	24.39%
TF-IDF	15.76%

Table 7: Accuracies for 3 methods using top 1000 features (tokenization Approach A)

Classic Accuracy	23.39%
Percent Contribution	22.70%
TF-IDF	20.67%

Table 8: Accuracies for 3 methods using top 1000 features (tokenization Approach B)

It should be kept in mind that this is a 10 class problem so random guessing would produce around 10% accuracy. TF-IDF actually exhibits worse errors rates because there are few classes so that many features appear in all classes and thus get ranked 0.

The actual accuracy is whether the features that were extracted make sense and can these features be used for finding relevant tweets that are of interest to the analyst. We have seen that the features identified distinguish classes. We saw for instance, that class c_9 carries features that are associated with more personal messages and class c_0 carries features that are closely associated with the Egypt uprising news topics (all other classes are somewhere in between). Going through the messages by hand in the classes we see that c_0 may contain tweets that should not be associated with class c_0, such as:

a) *"OMFGCould you believe it? My wife just purchased an Iphone for 42US$!!! http://moourl.com/5td4g Cairo #famouslies White Stripes DiPietro"*
b) *"WOW OMG JUST WOW -- search Twitter annnd Google side by side - http://bit.ly/hBxUBC #### #ifyouonlyknew Cairo Charles Barkley"*

Likewise other classes probably have tweets that have been misclassified. We use the top 100 features to reclassify the ten classes, but it is up to the analyst to determine if those features are enough (tweets that do not have the features are thrown away).

Among other things that might help in extracting useful tweets and increasing accuracy include analyzing how many times a tweet has been reposted (retweeted), how many people replied to it, and considering the geospatial component so that we focus only on tweets from a certain area. A way to filter irrelevant tweets would be to get rid of tweets that consist of features that are mentioned by less than N number of people (bottom up approach). This can be established through regression to determine a threshold. Another way is to try and understand most important features extracted and include the features from all of the tweets that mention those features (a top down approach). Filtering would get rid of spam and self-centered messages that do not give any insight in understanding the event of interest. In the next section we consider allowing the user to focus on features coming from a specific geo-location.

VI. Visualization

Tweets have a geospatial component to them so that they may be shown on a map. We have used JavaScript and Google Map API in order to visually present Twitter data for the Egypt dataset. The intention was to allow a user to click and drill down into tweets corresponding to some geographic location. In this way a user could perform analysis on how the features in one geographic location are different from tweets associated with a different geographic locations. The thought is that there will be more conversations in the local area where the event is actually taking place then in the rest of the world.

Based on the latitudes and longitudes in the Egypt dataset, we divide the world into a ten by ten grid. This grid serves effectively as a histogram and displays rings that correspond to number of tweets coming from a particular area (the center of the ring is the center for the particular cell on grid and so

some circles may appear on water, Figure 3). The Egyptian government limited Internet access so we actually see that most tweets that do have a geospatial location are from around the world (in particular from South Africa).

For the Egypt dataset, unfortunately only about 1% of data had a geo location associated with it, but this is typical as less than three percent of all tweets have geo-location information [28]. Here the user would select the Egypt province in order to focus on tweets from that area. The top 20 features would be used to filter tweets that don't have a geolocation in order to identify the next top 20 features. This iterative process discovers more and more tweets, avoids spam, and simplifies the computational requirements by not having to consider hundreds of thousands of tweets simultaneously. Results are still to follow in methods to appropriately use geographical information associated with tweets.

The benefit to using geo-location is that the user can focus on main features corresponding to the area of interest vs. discussions about the topic in neighboring regions. For example a victory in a sporting event will be discussed differently in the hometown vs. the rest of the country. The features coming from hometown will probably be positive about the hometown team. These features can then be used for finding towns that have similar feelings about the sport's team. This is an iterative process whereby more and more features can be discovered but at the root of those features will be the features associated with the hometown. The features can be listed in a hierarchical fashion and can be a means of organizing based on features and locations. Unfortunately only 1% of features have geo-locations, again results are still to follow for ranking using this approach.

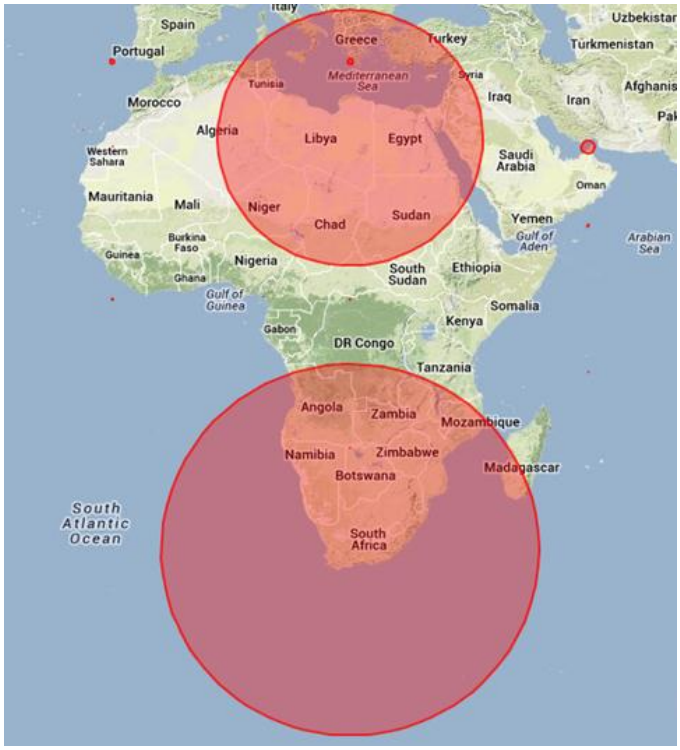


Figure 3. Selecting Tweets based on Geospatial Coordinates

VII. Discussion and Conclusions

In this paper, we explored the use of Twitter as a source of intelligence for determining the status of a pending, emerging, or on-going event. We believe Twitter can be queried for relevant data similar to queries performed on the World Wide Web. In order to make queries, an analyst should have a list of key features (query terms) related to some event of interest. In an attempt to extract those key features, we investigated a 10 pre-labeled class dataset by the ARL covering the Egypt uprising. The features from the labeled classes are used in a frequency matrix so that ranked features can be used to identify other relevant tweets (like Vector Space Model [VSM]). Once top features are identified, an analyst can use an open IE system to make queries for relevant tweets just as a person is searching for web documents on Google. These extra tweets are used to get at an even more robust feature set. Each loop generates a list of features that an analyst has to go through and approve. In this way, we foresee an iterative process between the analyst (feature approver), VSM (feature rankings), and an Open IE system (Twitter queries) in order to create catalogs of useful features for semantic analysis of activities. Catalogs of useful features can then be used for filtering and identifying events and activities of interest.

VIII. Acknoledgements

We thank Sue E. Kase and Liz Bowman at the Army Research Lab and Mike Hinman at the Air Force Research Lab for their help in getting access to the Egypt data.

References

[1] Manning, C. D., Raghavan, P., & Schutze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK.

[2] Salton, G., Wong, A., & Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18 (11), 613-620.

[3] Weaver, W. (1955). Translation. In Locke, W., & Booth, D. (Eds.), *Machine Translation of Languages*: Fourteen Essays. MIT Press, Cambridge, MA.

[4] Furnas, G. W., Landauer, T. K., Gomez, L. M., & Dumais, S. T. (1983). Statistical semantics: Analysis of the potential performance of keyword information systems. *Bell System Technical Journal*, 62 (6), 1753-1806.

[5] Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, Vol. 37, Issue 1, pp. 141-188.

[6] Rapp, R. (2003). Word sense discovery based on sense descriptor dissimilarity. In *Proceedings of the Ninth Machine Translation Summit*, pp. 315-322.

[7] Ritter, A., Clark, S., Mausam, and Etzioni, O. (2011) Named Entity Recognition in Tweets: An Experimental Study. *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, pp. 1524-1643.

[8] Finin, T., Murnane, W., Karandikar, A., Keller, N., Martineau, J., and Dredze, M. (2010). Annotating named entities in Twitter data with crowdsourcing. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

[9] Han, B. and Baldwin, T. (2011). Lexical normalization of short text messages: Makn sens a #twitter. In *49th Annual Meeting of the Association for Computational Linguistics*: *Human Language Technologies*, pp. 368-378.

[10] Gouws, S., Metzler, D., Cai, C., and Hovy, E. (2011). Contextual bearing on linguistic variation in social media. In *Proceedings of the Workshop on Language in Social Media*, pp. 20-29.

[11] Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (2011). Part-of-speech tagging for twitter: Annotation, features, and experiments. In *49th Annual Meeting of the Association for Computational Linguistics*: *Human Language Technologies*, pp. 42-47.

[12] Philips, L. (1990). Hanging on the Metaphone. *Computer Language*, 7(12), pp. 39-44.

[13] Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proc. of Association for Computational Linguistics*., pp. 384-394.

[14] Barbosa, L. and Feng, J. (2010). Robust sentiment detection on Twitter from biased and noisy data. In *Proc. of Conf. on Computational Logistics (COLING)*, pp. 36-44.

[15] Bollen, J., Pepe, A., & Mao, H. (2009). Modeling public mood and emotion: Twitter sentiment and socioeconomic phenomena. arXiv.org, arXiv:0911.1583v0911 [cs.CY] 0919 Nov 2009.

[16] Sakaki, T., Okazaki, M., and Matsuo, Y. (2010) Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pp. 851–860.

[17] Asur, S. and Huberman, B. A. (2010) Predicting the Future with Social Media. *Proc. IEEE/WIC/ACM Int'l Conf on Web Intelligence and Intelligent Agent Technology*, pp. 492-499.

[18] Etzioni, O., Fader, A., Christensen, J., Soderland, S., and Mausam. (2011). Open information extraction: the second generation. In *International Joint Conference on Artificial Intelligence*.

[19] Mausam, Schmitz, M., Soderland, S., Bart, R. and Etzioni, O. (2012) Open language learning for information extraction. In *Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 523-534.

[20] Gamallo, P. Garcia, M., and Fernandez-Lanza, S. (2012) Dependency-based open information extraction. In *Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP*, pp. 10-18.

[21] Beaumont, P. (2011) The truth about Twitter, Facebook, and the uprisings in the Arab world. *The Guardian* (Feb. 25, 2011); http://www.guardian.co.uk/world/2011/feb/25/twitter-facebook-uprisings-arab-libya

[12] Ghonim, W. (2011) Interviewed by Harry Smith. Wael Ghonim and Egypt's new age revolution. *60 Minutes* (Feb. 13, 2011); http://www.cbsnews.com/stories/2011/02/13/60minutes/main20031701.shtml?tag=contentMain;contentBody

[22] Al Jazeera. Timeline: Egypt's revolution (Feb. 14, 2011); http://english.aljazeera.net/news/middleeast/2011/01/201112515334871490.html

[24] Kase, S. E. and Bowman, L. ARL Egypt Twitter Data Set, 2012.

[25] Hongzhao Huang, Arkaitz Zubiaga, Heng Ji, Hongbo Deng, Dong Wang, Hieu Khac Le, Tarek Abdelzaher, Jiawei Han, Alice Leung, John Hancock and Clare Voss. 2012. Tweet Ranking based on Heterogeneous Networks. Proc. 24th International Conference on Computational Linguistics (COLING2012).

[26] Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into texts. In Proceedings of EMNLP, volume 4. Barcelona: ACL.

[27] Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28 (1), 11-21.

[28] Leetaru, K., Wang, S., Cao, G., Padmanabhan, A., and Shook, E. (2013). Mapping the global twitter heartbeat: The geography of twitter. *First Monday*, 18(5).

Situational Awareness from Social Media

Brian Ulicny, Jakub Moskal
VIStology, Inc
Framingham, MA
{bulicny,jmoskal}@vistology.com

Mieczyslaw M. Kokar
Northeastern University
Boston, MA
m.kokar@neu.edu

***Abstract*—This paper describes VIStology's HADRian system for semantically integrating disparate information sources into a common operational picture (COP) for humanitarian assistance/disaster relief (HADR) operations. Here the system is applied to the task of determining where unexploded or additional bombs were being reported via Twitter in the hours immediately after the Boston Marathon bombing in April, 2013. We provide an evaluation of the results and discuss future directions.**

Keywords—social media, situational awareness, Boston Marathon bombing.

I. INTRODUCTION

The Homeland Security Act (2002) defines situational awareness as "information gathered from a variety of sources that, when communicated to emergency managers and decision makers, can form the basis for incident management decision-making" [1]. Incident commanders for humanitarian assistance/disaster relief (HA/DR) operations are better able to understand a situation and make appropriate decisions if they can view all of the relevant information in an integrated common operational picture (COP) in a way that allows them to make sense of the situation without being overwhelmed with information. However, HA/DR commanders should not be expected to know where all the relevant information is stored or how it is encoded. It would be better if a system would identify how to meet a commander's high-level information needs on the basis of previously annotated information stores that could be brought to bear in an emergency. In such dynamic situations, it would be desirable, too, if the system allowed an administrator to quickly annotate new information stores in order to make them answerable to the commander's needs and, secondly, provide enough annotation that the system knew how to query, transform, load and analyze data relevant to the commander's high level needs into the system.

In a large-scale emergency situation, such as the aftermath of the Boston Marathon bombings on April 15, 2013 [2], masses of people communicated information rapidly via social media and react to those messages, shaping the situation. Some were reporting what they were observing on the scene; others were not on the scene and merely commented or relayed information they received from elsewhere. While often dismissed as trivial, FEMA officials have testified that, "Social media is imperative to emergency management because the public uses these communication tools regularly…. With one click of the mouse, or one swipe on their smartphone's screen, a message is capable of being spread to thousands of people and have a tangible impact" [3].

In order for a commander to understand the situation and respond effectively, the commander must therefore have access to what people are saying on social media, and this must be presented in such a way that the commander can respond to it effectively. However, neither the commander, nor his or her staff, has time to read all of those messages and identify what is relevant in order to assess the situation. Semantic machine processing of the messages must provide the necessary insight into the relevance of particular messages and summarize their significance to the commander's information needs in a way that enables decisions and actions.

VIStology's HADRian project, our internal name for an AFRL SBIR Phase II project titled "Fusion, Management, and Visualization Tools for Predictive Battlespace Awareness and Decision Making", is focused on being able to quickly integrate disparate data sources into a COP by semantically annotating datastores using an ontology against which commander queries can be issued to determine relevant repositories, formulate the proper query to issue to the repositories, extract results, reason with the query results, filter them and display them. This project extends previous data virtualization work at VIStology sponsored by the Office of Naval Research for representing and reasoning about maritime track repositories annotated with an ontology; the current project, sponsored by AFRL, includes entities of a variety of types for use in HA/DR situations. In this paper, we examine the application of this technology to deriving situational awareness from social media.

II. HADRIAN BACKGROUND AND CONCEPT OF OPERATIONS

In the first phase of this project, we developed techniques for dealing with a range of object types and a variety of data representation formats as well as a different type of interface (RESTful web services, GPS track servers, among others). A guiding principle in this project is that HA/DR commanders cannot dictate where relevant information is uploaded by users. Our goal is to make it usable wherever content creators upload it, as long as it is online. Thus, we need to develop techniques for accessing it in various ways. It turns out that RESTful Web Services are very common for retrieving information produced by 'ad hoc sensor networks' and so we

have focused on these. A proof-of-concept demo we developed reflects the retrieval and integration of information from disparate repositories into a single COP that are relevant to a scenario in which a plane crashes into a chemical factory. This scenario was drilled at Calamityville, a HA/DR training facility associated with the National Center for Medical Readiness at Wright State University on May 11, 2011. We used artifacts produced during this drill that exist in various repositories on the Web to illustrate our capabilities. We annotated the repositories that included them but do not modify the artifacts prior to incorporating them.

The Concept of Operations for our system is as follows:

1. A COP Administrator who manages the system **annotates repositories**, using an ontology, i.e. a formal representation of the conceptual domain.
2. The COP Administrator formulates High Level Query to **describe information needs** for current operation
3. The System **infers repositories** that may contain **relevant information** by **reasoning** over metadata that the repository has been **annotated** with.
    a. Information remains in place until it is needed. It is not initially all extracted, transformed and loaded (ETL).
    b. Users upload data wherever they usually upload it, not to a central repository.
4. The System issues appropriate **low level queries** to **repositories**
5. The System **filters out some** irrelevant data
6. The System **aggregates and displays data** in a COP
7. Users including the EOC (Emergency Operations Center) or Incident Commander and other operations center **interact with the data in the COP**.
8. The COP operator **pushes** elements of the displayed **information to users in the field** via their smartphone as needed.

In order to produce this demo, we developed:

1. Domain ontologies for representing repositories and queries, incorporating other ontologies as needed, such as UCore-SL [4] and a Distributed Interactive Simulation (DIS) Protocol Data Units (PDU) simulation data (for tracks) [11], to represent the conceptual and technical domain.

2. BaseVISor inference engine rules for reasoning about relevant repositories and rewriting query URLs in order to retrieve information elements from RESTful web interfaces and PDU sources that are relevant to this scenario. BaseVISor is VIStology's OWL 2 RL forward-chaining inference engine.

3. A novel technique for producing OWL representations of individual data items from the JSON output by RESTful web services. This allows us to generate OWL for reasoning without developing any custom software, on the basis of metadata and annotations alone.

4. Technology for integrating a variety of information types into the COP. We developed tools for integrating text, video, photos, and map overlays into a common COP based on Google Earth. We integrated Google Sketchup 3D facility models into the demo, and as well as GPS tracks, encoded as Distributed Interactive Simulation Protocol Data Unit binary data, as well as social media video, photos, and tweets in Phase I.

III. JIFX 13-4 FIELD EXPERIMENT

VIStology, Inc, recently conducted a field trial of its HADRian semantic information integration technology for Humanitarian Assistance/Disaster Relief operations at an invitation-only event sponsored by the Naval Postgraduate School held August 5-8, 2013, at McMillan Airfield, Camp Roberts, near Paso Robles, CA.

In the scenario that we pursued there, a commander needs to determine, on the basis of social media messages (here, only Twitter posts), where additional or unexploded bombs are being reported to be located (truly or falsely) in the aftermath of the Boston Marathon bombing in order to evaluate where to dispatch resources. In the immediate aftermath of the Marathon bombings, several locations were reported to have additional, unexploded bombs, all mistakenly as it turned out. Of course, it was not obvious at the time that the reports were false, and it was incumbent on public officials to maintain order and control at those sites if in fact they did contain a threat to public safety.

Our objective is to evaluate the feasibility of deriving situational awareness from a representative corpus of social media messages gathered immediately after the Boston Marathon bombing. The corpus consists of approximately 0.5 million messages that span the three hours following the bombing. In this experiment, information from social media users (here, Twitter users) was analyzed for answers to the high level query "Where are people reporting that additional or unexploded bombs have been found?"[1] Answers to this question were identified and presented in the COP in an appropriate way. The information included represented the following:

Where are additional/unexploded bombs being reported to exist?;

When were those messages propagated?;

How often have these messages been propagated (i.e. the amount of attention being directed to each location)?;

We were not able yet to represent, a future goal, answers to:

[1] This scenario was suggested to us by Desi Matel-Anderson, FEMA Innovation Advisor and Think Tank Strategic Vision Coordinator, at RELIEF 13-3.

How reliable and credible are the reports of a bomb at that location.

IV. SYSTEM DESCRIPTION

The HADRian system can be thought of as having four functionalities that are relevant to this scenario:

A. Query Formulation and Repository Annotation
B. Relevance Reasoning and Repository Querying
C. Results Reasoning
D. Interactive Display

A. Query Formulation and Repository Annotation.

High level information needs are represented in our system ontology as instances of an OWL class called High Level Query (HLQ). In our system, an HLQ is not a query string in any particular query language, such as SQL or SPARQL. Rather, it is a description of one or more such queries, represented in OWL. That is, it should be possible to derive the OWL description of a query string by parsing and analyzing the query. We have made some attempts at translating SPARQL queries and even natural language queries into their OWL descriptions, automatically. However, at present, we rely on manually encoding HLQs in OWL directly.

A High Level Query is assigned various 'scopes' in the ontology: a Region Scope, a Time Scope, a Topic Scope, a Thing Scope and a Source Scope. Some of these scopes are related via annotation properties to classes or individuals in the ontology (in the case of Thing and Topic Scopes). An HLQ is related via an object property to individuals in the case of Time and Region Scopes. An HLQ essentially corresponds to an instance of a query of the form:

Find all instances of class T produced by instances of class S that are about instances of class U that existed in region R during temporal period P

Here, class T corresponds to the Thing Scope of the HLQ. A Thing Scope relates a query to the kind of thing that constitutes an answer to the query. For example, in English, "who" queries seek a Person or subclass of Person as an answer (e.g. Q: "Who can sign my timecard?" A: "Bill", "a manager"). A Topic Scope specifies what the specified 'things' from the Thing Scope are about: e.g. magazines about *Sports*. In the query template above, R corresponds to the Region Scope, which is an individual region in the ontology. P corresponds to the Time Scope, which is an individual temporal range in the ontology. The Source Scope S indicates that all of the things that satisfy the query must have been produced by an individual of class S or a subclass of S. The classes that are represented may be expressed with arbitrarily complex OWL class expressions.

Repositories are also a class in our ontology. Every repository also has a Thing, Topic, Region, Time and Source Scope. Thus, for example, a repository of tweets about traffic accidents in Paso Robles, CA, during 2012 from the Paso Robles (CA) Police Department would have the following scopes:

Thing Scope: StatusUpdate
Topic Scope: TrafficAccident
Region Scope: Paso Robles, CA
Time Scope: 2012
SourceScope: Paso Robles Police Department

HLQs and Repository Annotations are represented in an OWL ontology that incorporates the UCore-SL ontology [4] and aspects of the Dublin Core [5] and Geonames ontologies [6].

Any ontology editor can be used to annotate repositories and formulate queries. We currently use Protégé 4.x for this purpose, but any other OWL editor would do.

B. Relevance Reasoning and Repository Querying

Relevance Reasoning, in our system, is the process of identifying which repositories are relevant to a High Level Query based on its OWL annotations [8]. In HADRian, we do not examine the contents of the repository in identifying a relevant repository. The system only considers the metadata that has been assigned to it.

A Repository is inferred to be relevant to a HLQ if (but not only if) its scopes overlap with the Thing, Topic, Region and Time scopes of the HLQ. If a scope is specified in terms of a class, then a subclass or superclass overlaps with it. Regional and temporal overlaps are defined in the obvious way. A Topic Scope defined in terms of an individual coincides with any coreferential term.

A Repository, in our system, is a collection of items that could be represented in the COP. Repositories are a collection of items, and as such, they may be defined *extensionally* as pre-specified collection of things or *intentionally* as items that satisfy certain criteria, expressed as a query to a larger repository. For example, a collection of photos in some individual user's Flickr online photo album (flickr.com) represents a collection defined extensionally: the collection was defined by the user's selection of photos for that album. A Flickr query for photos taken in Yosemite Park on a particular date, however, is a repository that is determined *intensionally*. The set of photos that meet this criterion is not necessarily known in advance.

Each Repository must have a URL associated with it that enables the system to retrieve (extensional) or query (intensional) the data. Many of the repositories we deal with have RESTful interfaces. A query-defined repository for a RESTful interface may have parameters that are specified at run time based on the High Level Query. For example, a query for businesses listed in Yelp (yelp.com) may have a parameter for a zipcode that is filled at runtime by the zipcode corresponding to the area(s) that is (are) in the Region Scope of the HLQ.

For the Boston Marathon scenario, the HLQ has obvious Region (Boston, MA) and Time (April 15, 2013) scopes, but the Thing and Topic Scopes are not as obvious. The Thing Scope of the HLQ is defined as the class GeoFeaturesMentionedInStatusUpdates. This class is defined

as a subclass of the intersection of the classes GeographicFeature (a UCore-SL class defined as "A PhysicalEntity whose (relatively) stable location in some GeospatialRegion can be described by location-specific data.") and the class of things are the subject of the mentionedIn object property with respect to some StatusUpdate. The class StatusUpdate is equivalent to the sioc:Post class, defined as "An article or message that can be posted to a Forum" [2].

The repository of tweets in this scenario thus has the Thing Scope StatusUpdate, but the HLQ has a Thing Scope of GeoFeaturesMentionedInStatusUpdates, which is neither a super- nor subclass of StatusUpdate. Therefore, it is not within the Thing Scope of the HLQ. A relevance reasoning rule, specified in BaseVISor rule language, states that if an HLQ has a Thing Scope that is a subclass of things mentionedIn some class C and a repository has a Thing Scope that is a subclass of C, then the repository is relevant to the HLQ.

BaseVISor is VIStology's customizable, forward-chaining OWL 2 RL inference engine. BaseVISor (vistology.com/basevisor) provides inference rules for the OWL 2 RL language profile, but it can be extended with custom rules. These rules may be augmented with user-supplied procedural attachments that perform custom functions in addition to default functionality for mathematical functions, string operations and the like [7].

In this case, the repository of tweets is pre-existent. Therefore, it is extensionally defined and does not require any run-time instantiation of lower level query parameters. We simply extract the contents of the repository and convert them to OWL, in order to do results reasoning.

The Topic Scope of the HLQ and the Repository both consist of the individual BostonMarathon2013 and the class UnexplodedBombs. Not every tweet in the repository is about UnexplodedBombs, although they are all presumed to be about the 2013 Boston Marathon. The class UnexplodedBombs is associated with a regular expression in the ontology that allows us to filter the query contents to only those tweets that are about both subjects.

C. Results Reasoning

After the relevant tweets are converted to OWL using a template that is part of the metadata annotation of the repository, BaseVISor is again used to reason about the results, in order to extract the required elements. Here a set of custom BaseVISor rules is used to identify locations mentioned in tweets about both unexploded bombs and the 2013 Boston Marathon. These rules produce a set of phrases that refer to locations. These location phrases are then mapped to known locations using a heuristic algorithm that chooses among the results of querying the Google Places and Google Maps Geocoding APIs, using the location phrase and a geographic region corresponding to Boston as the parameters of the search. This process associates locatable phrases with known locations and removes some phrases that are syntactically plausible but for which no identifiable location can be associated. For example, one of the extracted location phrases is 'BPD Commissioner Ed Davis', based on its context. This phrase corresponds to no known place by querying the Google APIs, so it is dropped from the output. Location phrases that do result in known places are collated. Several extracted phrases may coincide with the same known place, according to one or more of the Google APIs. A count of the number of tweets that are associated with each known place is kept. Various metadata elements associated with the known place are inserted into the KML document that is displayed as the result of the query.

D. Interactive Display

Finally, the KML is displayed in the COP as an answer to the High Level Query. Each placemark is labeled with one of the location phrases that produced it. A number in parentheses next to the placemark's title indicates the number of tweets that mentioned one of the location phrases mapping to this location. We emphasize this fact by rendering polygons underneath the placemarks that also correspond to the location volume in tweets: the higher and darker the color, the more frequently mentioned was the location. Clicking on the placemark reveals the phrases that produced the placemark, the type of place (according to Google), and the API source (Figure 1).

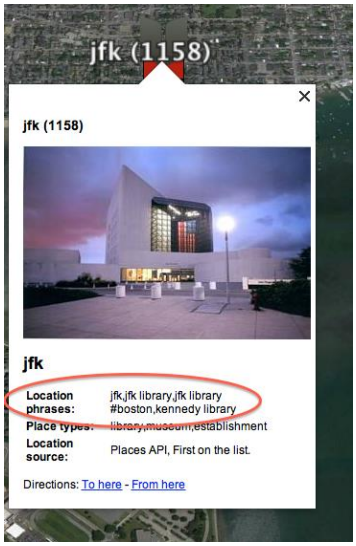


Figure 1 Expanded placemark shows location phrases that resulted in the placemark, number of tweets (1158), the type of place (library, museum) and the API source.

Each placemark can be removed from the COP by unchecking a widget in the list of placemarks on the left hand side of the COP (Figure 2). This set of placemarks can be viewed alongside other layers in Google Earth, such as baselayers presenting a photographic map of the various structures in the region as well as street names and other geographic features and attributes.

[2] Semantically-Interlinked Online Communities (sioc-project.org)

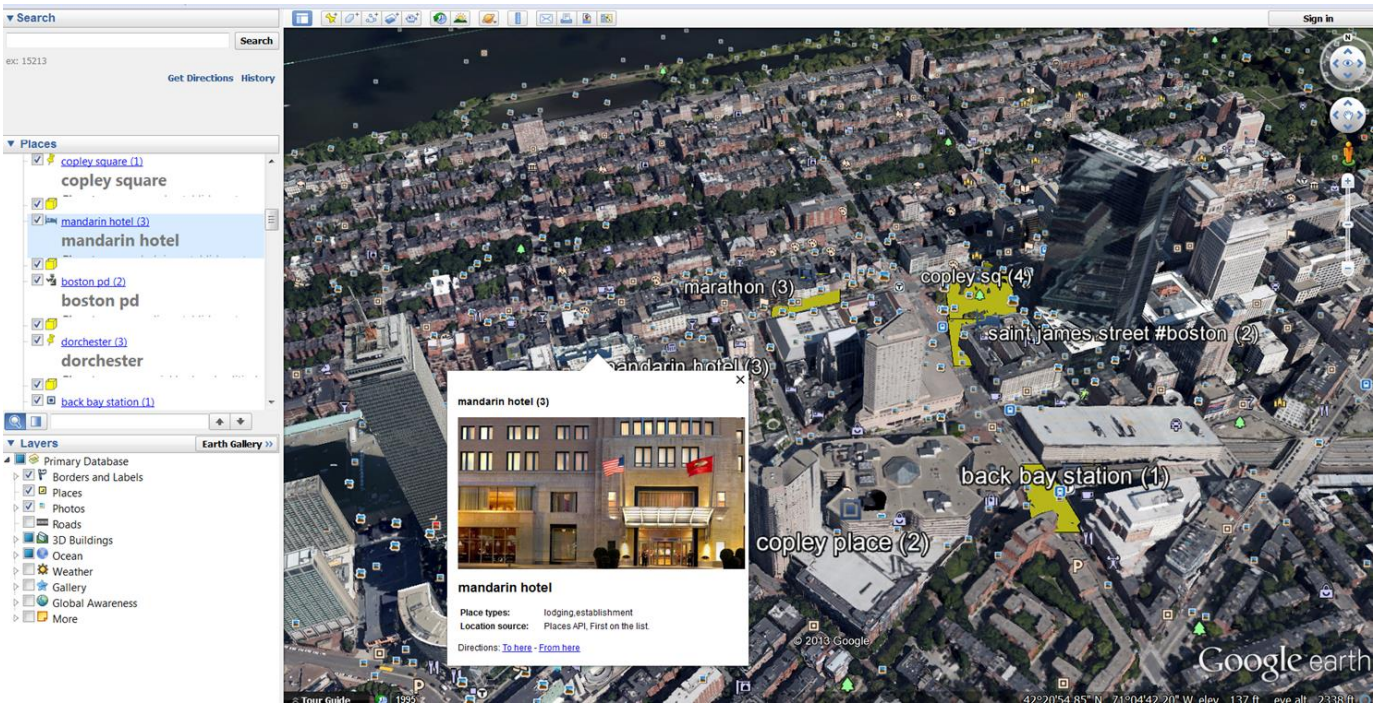


Figure 2 COP Indicating that three tweets about unexploded bombs mention the Mandarin Hotel, four mention Copley Square, one Back Bay Station and so on.

V. EVALUATION

In this exercise, we annotated a repository containing 509,795 twitter messages containing the hashtag #bostonmarathon between 4:06 PM and 7:04 PM on April 15, 2013, retrieved using Twitter APIs. The bombs are said to have exploded at 2:49 PM that day. The corpus was collected by Andrew Bauer and his colleagues at Syracuse University's School of Information Studies's NEXIS lab and made available on the Web as a CSV file.[3] The file contains the tweet ID number, text, creation time, associated latitude/longitude (if there is one) and user ID.

The latitude and longitude in the file represents the location of where the user sends the tweet from, not necessarily the location about which the user is reporting. Only 8,300 of the tweets had geocoded origins, or about 1.6% of the corpus. Generally, less than 1% of twitter users have enabled geotagging their locations using the location services on their smartphones or other devices [9][10]. In disaster relief datasets that we have examined, geotagged tweets approach 2% of the corpus. We were not concerned with the source location of tweets, but locations that were mentioned in the tweets, so we ignored these fields even when they were non-null. The repository was annotated in our ontology as described above.

We evaluated our processing by evaluating: the recall and precision of identifying tweets that mentioned unexploded bombs and the like; the recall and precision of identifying phrases specifying a location in the tweets; and the precision of associating a location phrase with a known place, using the Google APIs mentioned previously.

Precision in automatically identifying instances of a category is the ratio of true, positive identifications to positive identifications. Recall is the ratio of true, positive identifications to positive instances in the corpus as a whole. Finally, the F1-measure characterizes the accuracy of a categorization task as a whole by combining the recall and precision into a single metric, weighing each equally:

$$F_1 = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

To begin with, we did not evaluate the precision and recall of categorizing the corpus with respect to the topic of the Boston Marathon. We assume that all of the tweets in the corpus were about the 2013 Boston Marathon because of the time period in which they were sent in temporal proximity to the bombings. It is possible that some of the tweets in the corpus contain the hashtag #bostonmarathon but are in some sense not about the 2013 Boston Marathon. We have no way to evaluate the recall of this corpus. That is, we have no way to evaluate how many tweets were sent that were about the 2013 Boston Marathon but that did not contain this hashtag and were not collected in this corpus.

Of the tweets in this corpus, we identified 7,748 tweets that were about additional or unexploded bombs with a precision of 94.5%, based on a random sample of 200 tweets identified as such. That is, only 1.5% of the original corpus was identified as referring to additional bombs, using our pattern matching.

[3] https://www.dropbox.com/s/h8wezi2y6pzqfh4/041513_1606-1704_tweets.zip

Based on a random sample of 236 tweets from the original corpus, our recall (identification of tweets that discussed additional bombs) was determined to be 50%. That is, there were many more ways to refer to additional bombs than our rules considered. Thus, our F1 measure for accurately identifying tweets about additional bombs was 65%. Nevertheless, because of the volume of tweets, this did not affect the results appreciably.

Having thus reduced the corpus 98.5% in this way to only tweets that discussed unexploded bombs in addition to referring to the 2013 Boston Marathon, we now evaluate the precision and accuracy of identify location phrases. Location phrases were identified purely by means of generic pattern matching. We did not use any list of known places. Nor did we include any scenario-specific patterns. The precision with which we identified location phrases was 95%. That is, in 95% of the cases, when we identified a phrase as a location phrase, it actually did refer to a location in that context. Mistakes included temporal references and references to online sites. Our recall was only 51.3% if we counted uses of #BostonMarathon that were locative. (We mishandled hashtags with camel case.) Alternatively, since all of the tweets contained some variant of the hashtag #bostonmarathon, this is a somewhat uninformative location phrase. If we ignore this hashtag, then our recall was 79.2%. That is, of all the locations mentioned in tweets about additional bombs at the Boston Marathon, we identified 79.2% percent of the locations that were mentioned. Using the more lenient standard, our F1 measure for identifying location phrases in the text was 86.3%.

Our precision in associating tweets with known places via the Google APIs was 97.2%. Our precision in assigning unique location phrases to known places via Google APIs was 50%. That is, there were many location phrases that were repeated several times that we assigned correctly to a known place, but half of the unique phrase names that we extracted were not assigned correctly. Ten location phrases that were extracted corresponded to no known locations identified via the Google APIs. These included location phrases such as "#jfklibrary" and "BPD Commissioner Ed Davis". The former is a phrase we would like to geolocate, but lowercase hashtags which concatenate several words are challenging. The latter is the sort of phrase that we expect would be rejected as non-geolocatable. See Table 1.

Table 1 Top 20 Identified Places with Number of Tweets

Known Place	#Tweets
JFK Library	1158
Boston	629
Boston Marathon	325
St Ignatius Catholic Church	47
PD	29
Boylston	8
CNN	5
Copley Sq	4
Huntington Ave	4
Iraq	3
Mandarin Hotel	3
Dorchester	3
Marathon	3
US Intelligence	3
Copley Place	2
Boston PD	2
BBC	2
Cambridge	2
John	2
St James Street #Boston	2

More qualitatively, the Twitter processing we described here resulted in 38 ranked places on the COP that were associated with additional or unexploded bombs. We compared these places with the places that were mentioned in the live blogs that were set up by CNN[4], the New York Times[5] and the Boston Globe[6] immediately following the bombings. These blog sites mentioned the following locations (only once, each)

Location [Source]: (# of Tweets Identified with That Location)

Boylston Street [Globe, CNN]: 8
Commonwealth Ave near Centre Street, Newton [Globe]: 0
Commonwealth Ave (Boston) [Globe]: 0
Copley Square [NYT]: 4
Harvard MBTA station [Globe]: 0
JFK Library [CNN, Globe, NYT]: 1158
Mass. General Hospital [Globe, NYT]: 0
(glass footbridge over) Huntington Ave near Copley place [Globe]: 4
Tufts New England Medical Center [NYT]: 0
Washington Square, Brookline [NYT]: 0

For three of these sites – Mass. General Hospital, Tufts Medical Center and Washington Square, Brookline, reports of unexploded bombs or suspicious packages occurred after the end of the tweet collection period, at 7:06 PM. Otherwise, the recall of our system was good, missing only the report of unexploded bombs at the Harvard MBTA station. A few tweets mentioning such a threat were in our corpus, but the

[4] http://news.blogs.cnn.com/2013/04/15/explosions-near-finish-of-boston-marathon/comment-page-18/
[5] http://thelede.blogs.nytimes.com/2013/04/15/live-updates-explosion-at-boston-marathon/
[6] http://live.boston.com/Event/Live_blog_Explosion_in_Copley_Square?Page=16

system failed to pick them up, either due to capitalization issues or unexpected use of hashtags.

Additionally, on average, tweets reflecting these locations were produced 11 minutes prior to their being reported on the sites mentioned. Thus, the tweet processing was more timely and more comprehensive than simply relying on a handful of news sites alone for situational awareness

I. CONCLUSION

In this paper, we described a system for integrating disparate information sources into a COP for Humanitarian Assistance/Disaster Relief operations by means of semantic annotations and queries, using a common ontology. We described the operation of the system and evaluated the results of an experiment in annotating and querying social media data streams in order to produce situational awareness. We applied our technology to a repository of tweets collected in the immediate aftermath of the Boston Marathon bombings in April, 2013, and demonstrated that a ranked set of places could be incorporated into the COP, showing the prominence of each site by tweet volume that was reported as being the site of an additional unexploded bomb or bombs. We evaluated the results formally and compared the results with the situational awareness that could be gleaned only from mainstream media blogs being updated at the same time. On average, the automatic processing would have had access to locations from tweets eleven minutes before these sites were mentioned on the mainstream media blogs. Additionally, sites that were prominent on Twitter (e.g. St Ignatius Church at Boston College or the Mandarin Oriental Hotel in Boston) were not mentioned on the news blog sites at all. We believe that these results show that this approach is a promising one for deriving situational awareness from social media going forward.

ACKNOWLEDGMENT

This work was performed under AFRL contract FA8650-13-C-6381 “Fusion, Management, and Visualization Tools for Predictive Battlespace Awareness and Decision Making”. Thanks also to the JIFX 13-4 participants for helpful feedback.

REFERENCES

[1] United States Code, 2010 Edition Title 6 - DOMESTIC SECURITY CHAPTER 1 - HOMELAND SECURITY ORGANIZATION SUBCHAPTER V - NATIONAL EMERGENCY MANAGEMENT Sec. 321d - National Operations Center 6 U.S.C. §321d(a)

[2] FEMA. Lessons Learned - Boston Marathon Bombings: The Positive Effects of Planning and Preparation on Response. August 2, 2013.

[3] Shayne Adamski. Written testimony of FEMA for a House Homeland Security Subcommittee on Emergency Preparedness, Response, and Communications hearing titled “Emergency MGMT 2.0: How #SocialMedia & New Tech are Transforming Preparedness, Response, & Recovery …” July 9, 2013

[4] Barry Smith, Lowell Vizenor and James Schoening, “Universal Core Semantic Layer“, Ontology for the Intelligence Community, Proceedings of the Third OIC Conference, George Mason University, Fairfax, VA, October 2009, CEUR Workshop Proceedings, vol. 555.

[5] Dublin Core Metadata Initiative, http://dublincore.org

[6] GeoNames Ontology, http://www.geonames.org/ontology/

[7] C. Matheus, B. Dionne, D. Parent, K. Baclawski and M. Kokar. BaseVISor: A Forward-Chaining Inference Engine Optimized for RDF/OWL Triples. In Digital Proceedings of the 5th International Semantic Web Conference, ISWC 2006, Athens, GA, Nov. 2006.

[8] M. Kokar, B. Ulicny, and J. Moskal. Ontological structures for higher levels of distributed fusion in Distributed Data Fusion for Network-Centric Operations, D. Hall, C.-Y. Chong, J. Llinas, and M. Liggins II, Eds., ed: CRC Press, 2012, pp. 329-347.

[9] Cheng, Z.; Caverlee, J.; and Lee, K. 2010. You are where you tweet: A content-based approach to geo-locating twitter users. In CIKM ’10.

[10] LEETARU, Kalev et al. Mapping the global Twitter heartbeat: The geography of Twitter. First Monday, [S.l.], apr. 2013. ISSN 13960466. Available at: <http://firstmonday.org/ojs/index.php/fm/article/view/4366>. Date accessed: 04 Sep. 2013. doi:10.5210/fm.v18i5.4366.

[11] McGregor, D., Brutzman, D., Armold, A., and Blais, C. L., "DIS-XML: Moving DIS to Open Data Exchange Standards," Paper 06S-SIW-132, Simulation Interoperability Standards Organization, 2006 Spring Simulation Interoperability Workshop, Huntsville, AL, April 2006

Towards a Cognitive System for Decision Support in Cyber Operations

Alessandro Oltramari and Christian Lebiere
Functional Modeling Systems Lab
Department of Psychology
Carnegie Mellon University
Pittsburgh, USA

Lowell Vizenor
Refinery 29
New York, NY, USA

Wen Zhu
Alion Science and Technology
Washington D.C., USA

Randall Dipert
Department of Philosophy
University of Buffalo

Abstract— **This paper presents the general requirements to build a "cognitive system for decision support", capable of simulating defensive and offensive cyber operations. We aim to identify the key processes that mediate interactions between defenders, adversaries and the public, focusing on cognitive and ontological factors. We describe a controlled experimental phase where the system performance is assessed on a multi-purpose environment, which is a critical step towards enhancing situational awareness in cyber warfare.**

Keywords—ontology, cognitive architecture, cyber security

I. INTRODUCTION

A cyber attack by a hostile nation-state or political organization is widely regarded as one of the most serious threats that the U.S. will face in the next decades. While greatly increased use of information systems has contributed enormously to economic growth, and has fueled a much more efficient and agile national defense, it has also made the U.S. enormously vulnerable to a variety of Internet and non-Internet cyber attacks, and to cyber espionage [1].

There are numerous factors that make cyber warfare and pure cyber defense, namely cyber security, especially problematic. The kinds of threats are diverse: destruction or theft of data, or interference with information systems and networks, across a spectrum of private and public interests. The legal and ethical status of cyber attacks or counterattacks by states are also unclear, at least when deaths or permanent destruction of physical objects does not result. It is still an open question what U.S. policy is or should be, and how cyber threats are analogous to traditional threats and policies—for example whether "first use" deterrence, and in-kind responses apply, and whether a policy of pure cyber defense does not put the far greater burden on attacked rather than attacking nations [2]. As this overview may suggest, untangling the complexity of cyber attacks becomes a key element for augmenting situational awareness in the cyber environment: in this position paper, we propose to tackle this problem from a semantic and cognitive modeling perspective, combining ontologies and cognitive architectures into an intelligent system capable of supporting humans in cyber operations as wells as acting autonomously as a team member.

The paper is divided into four main parts. After introducing some aspects of special interest to modeling cyber warfare (Section II), in Section III we present a hybrid decision support system based on cognitive architectures and ontologies. Section IV unfolds the experimentation plan to test the system by means of a scalable synthetic environment, and Section V delineates a framework of implementation centered on an object-based infrastructure.

II. RELEVANT CHARACTERISTICS OF CYBER WARFARE

In general, time variables play an important role in the design of decision support systems [3]: temporal constraints become even more stringent when those systems have to deal with cyber attacks, where real time responses are typically hindered by the knowledge-intensive nature of cyber operations and associated tasks. Some decisions on where and when to invoke various methods of cyber defense and mitigate damage, as well as decisions to launch a cyber counterattack, need to be made quickly. Large-scale cyber attacks or counterattacks are likely going to require careful, human decision-making for some time into the future. Yet there are other responses to cyber attacks or cyber espionage that could and should be done immediately, such as revoking an employee's access if suspicious activity is detected, blocking all remote access or from certain URLs and through certain servers, immediate assessment of likely damage and risks, and so on. What we propose in this paper is the building of a cognitive system for decision support that will emulate ideal human responses to cyber attacks. This would be accomplished through careful design of its architecture, both in terms of cognitive mechanisms and knowledge resources, and by comparing its outputs on case studies with actions of human agents. The benefits are threefold. First, by cognitive modeling we come to better understand the mechanisms underlying human decisions in the realm of cyber warfare and cyber

This research was partially supported by a Defense Threat Reduction Agency (DTRA) grant number: HDTRA1-09-1-0053 to Christian Lebiere (Principal Investigator) and Alessandro Oltramari.

espionage, coupling the cognitive aspects and the semantic contents of decision-making. Second, after extensive testing we could use this intelligent decision-making system to recommend steps to human decision-makers—e.g., recommendations to gather further information, or actually to act in a certain way and to assess the risks of not acting. Finally, in cases where the reliability of the system is high, and where time is of the essence or the actions have little risk (such as revoking one employee's system access, or access to one URL), the intelligent system could act swiftly and autonomously.

Some forms of attacks, such as Distributed Denial Of Service (DDoS) and other botnet jamming of networks or servers, show signs of admitting purely technological solutions. However, human error by employees has repeatedly been cited as the most common source of vulnerability [4], [5], [6], [7]. One technique of gaining illegitimate access to an information system that still appears with remarkable frequency is spear-phishing: emails to DOD employees or defense contractors with spoofed addresses from acquaintances that seem to have a harmless photograph, PDF, or other attachment[1]. While this exploitation might not alone gain direct access to secure systems, it may allow an attacker to gather personal information that can be used to guess passwords, answer security questions, and so on. Social networking sites and other open data and the use of analytics allow attackers to identify employers, friends, relatives, shopping and driving habits, and so on. This aids an attacker enormously in the identification of targets and gaining access: for instance, in a recent case the New York Times' sites were brought down when a group claiming to be the Syrian Free Electronic Army used social media and spear phishing to gain access to employees' passwords to the server that handled the NY Times' Domain Network System (DNS). Likewise even if smartphones and other portable devices are not used at secure locations and do not contain classified or sensitive data, hacking into them (or intercepting cellular and WiFi communications, including with vehicles and home monitoring devices) can provide personal data that can be utilized to make direct attacks.

III. Towards a Cognitive System for Decision Support in Cyber Warfare

A. General methodology

Our approach is inspired by the notion of "sociotechnical system" [8], which emphasizes the interaction between people and technology in workplace. Ontology analysis has recently proved to be an effective tool for investigating these complex aspects [9]: nevertheless, the interactive nature of socio-technical systems demands a broader framework, where human behavior can be studied not only in terms of action schematics, planning and rules, but also as a genuinely cognitive phenomenon, which can be properly investigated only as a dynamic system. Accordingly, the key elements of our proposed method for modeling cyber operations are:

- Cognitive architecture – design and development of cognitive models of decision-making in cyber defense based on ACT-R[2] cognitive architecture [10]. The models will focus on: learning mechanisms, memory and attentional limitations, decision-making strategies, risk perception, and trusted judgments.

- Ontologies – design and development of applied formal ontologies to 1) serve as a knowledge base for our cognitive models (*Cyber Security Ontologies*) and to 2) classify and annotate cyber security test and training data (*Scenario Ontologies*).

- Live, Virtual, Constructive (LVC) Integration – Enable the analysis of cyber defense strategies; support training for cyber security personnel; validate the cognitive models developed with an attack/mitigate/counter-attack scenarios and enhance them by leveraging learning mechanisms.

By integrating these elements in a coherent multi-purpose system, we aim at unraveling the complex structures that mediate interactions among defenders, adversaries and the public: in this respect, the overall goal is to enhance situational awareness in cyber warfare by assessing human performance in a simulated environment. The system is also meant to interact autonomously in a *hybrid* team, i.e. playing the role of a "teammate" sentinel in support of humans, eventually capable of prompting decisions and perform actions in more mature stages of development.

To provide a richer characterization of our approach, Section *B* illustrates the functional requirements of the envisioned system, while Section *C* and *D* will narrow the focus to, respectively, ACT-R cognitive architecture (the central component of the system) and the ontologies needed to frame the knowledge component of the architecture.

B. Functional models of cyber operations

Modeling decision-making in the cyber security framework requires multiple factors to be investigated: (*i*) the size and the variety of *knowledge* which is necessary to classify and analyze attacks and defensive actions; (*ii*) the *flexible behavior* required by coupling alternative strategies of response to specific cyber threats, updating and revising strategies when the circumstances of the attack or the environmental conditions evolve; (*iii*) *learning by experience* how to deal with cyber attacks; (*iv*) *interacting* in a team by building a mental representation of the co-workers as well as of the enemies. These factors can be mapped to the 12 criteria distilled in [11] (from the original list compiled by Newell in [12]) that a cognitive architecture would have to satisfy in order to achieve human-level functionality. In these regards, cognition is not considered as a "tool" for optimal problem solving but, rather, as a set of limited information processing capacities (so-called 'bounded rationality' [13])[3]. In a similar fashion, Wooldridge had identified the requirements that an agent should satisfy in order to act on a rational basis [14], namely: *reactivity*, the capacity of properly reacting to perceptual stimuli; *proactivity*, the capacity of operating to pursue a goal; *autonomy*, implying

[1] Because of their prevalence and complexity in terms of kind and number of cognitive agents, we intend to include these as paradigms of our use-cases.

[2] Pronounced, "act-ARE": Adaptive Control of Thought—Rational.

[3] Despite the relevance of emotions in decision-making [34], our approach doesn't extend to the investigation of affective aspects at this stage.

an unsupervised decision making process; *social ability*, the capacity of interacting with other agents and revising mental states accordingly.

State-of-the-art research on cognitive architectures (SOAR, ACT-R, CLARION, OpenCog, LIDA, etc.) has produced a significant amount of results on specifying this extensive range of functions[4]: by and large, ACT-R has accounted for the broadest range of cognitive activities at a high level of fidelity, reproducing aspects of human data such as learning, errors, latencies, eye movements and patterns of brain activity [10]. However, these results have often involved relatively narrow and predictable tasks. Most importantly, cognitive architectures have just started to tackle the problem of how to model *social ability* [15], which is a crucial aspect of our approach. A fundamental feature of human social ability is "mindreading" [16], i.e. to understand and predict the actions of others by means of postulating their intentions, goals and expectations: this process of interpretation is feasible only if an agent can learn to *represent* the mental states of others on the basis of cumulative experience and background knowledge, combining the resulting mental model with the continuous stream of data from the environment, aiming at replicating the cognitive processes that have likely motivated the other agents to perform the observed actions. Scaling up ACT-R to account for more extensive multi-agent scenarios can help to build comprehensive models[5] of social conflict and cooperation, which are critical to discern the governing dynamics of cyber defense. But if leveraging the ACT-R framework might be sufficient to replicate the *mechanisms* described in (*ii*)-(*iv*), the knowledge functionality (*i*) can to be fulfilled only by injecting a fair amount of highly expressive knowledge structures into the architecture: accordingly, ontologies can be provide these structures in the form of semantic specifications of declarative memory *contents* [17]. As [18], [19], and [20] show, up to this time most research efforts have focused on designing methods for mapping large knowledge bases to ACT-R declarative module, but with scarce success. Here we commit to a more efficient approach: modular ontologies. Modularity has become a key issue in ontology engineering. Research into aspects of ontological modularity covers a wide spectrum: [21] gives a good overview of the breadth of this field. Our modular approach guarantees wide coverage and "manageability": instead of tying ACT-R to a single large ontology, which is hard to maintain, update and query, we propose a *suite of* ontologies that reliably combine different dimensions of the cyber defense context, e.g. representation of secure information systems at different levels of granularity (requirements, guidelines, functions, implementation steps); categorization of attacks, viruses, malware, worms, bots; descriptions of defense strategies; the mental attitudes of the assailant, and so on.

In our context, the computational system resulting from the combination of cognitive and knowledge functionalities aims at fostering a better understanding of cyber attacks, supporting human operators in cyber warfare, eventually cooperating with them in well-defined synthetic environments. The rest of the paper presents in more detail the basic components of such a hybrid framework.

C. Replicating cognitive mechanisms with ACT-R

Cognitive architectures attempt to capture at the computational level the invariant mechanisms of human cognition, including those underlying the functions of control, learning, memory, adaptivity, perception, decision-making, and action. ACT-R [10] is a modular architecture including perceptual, motor and declarative memory components, synchronized by a procedural module through limited capacity buffers (see figure 1 for the general diagram of the architecture). Declarative memory module (DM) plays an important role in the ACT-R system. At the symbolic level, ACT-R agents perform two major operations on DM: 1) accumulating knowledge "chunks" learned from internal operations or from interacting with objects and other agents populating the environment and 2) retrieving chunks that provide needed information. ACT-R distinguishes 'declarative knowledge' from 'procedural knowledge', the latter being conceived as a set of procedures (production rules or "productions") which coordinate information processing between its various modules [10]: according to this framework, agents accomplish their goals on the basis of declarative representations elaborated through procedural steps (in the form of *if-then* clauses). This dissociation between declarative and procedural knowledge is grounded in experimental cognitive psychology; major studies in cognitive neuroscience also indicate a specific role of the hippocampus in "forming permanent declarative memories" and of the basal ganglia in production processes (see [22], pp. 96-99, for a general mapping of ACT-R modules and buffers to brain areas and [23] for a detailed neural model of the basal ganglia's role in controlling information flow between cortical regions). ACT-R performs cognitive tasks by combining rules and knowledge: for reasons of space, a complete analysis of how the architecture instantiates this cognitive-based processing is not suitable here. Nevertheless, two core mechanisms need to be mentioned: *i) partial matching,* the probability of association between two distinct declarative knowledge chunks, computed on the basis of adequate similarity measures (e.g. a bag is more likely to resemble a basket than a tree); *ii) spreading of activation*, the phenomenon by which a chunk distributionally activates the different contexts in which it occurs (a bag can evoke shopping, travel, work, etc.). These two basic mechanisms belong to the general sub-symbolic computation underlying chunk activation, which in ACT-R controls the retrieval of declarative knowledge elements by procedural rules. In particular, ACT-R chunk activation is calculated by the following equation:

$$A_i = \ln \sum_j t_j^{-d} + \sum_k W_k S_{ki} + \sum_l MP_l Sim_{li} + N(0,\sigma) \qquad (1)$$

On the basis of the first term, the more recently and frequently a chunk i has been retrieved, the higher the activation and the chances of being retrieved (t_j is the time elapsed since the j^{th} reference to chunk i and d represents the memory decay rate). In the second term of the equation, the contextual activation of a chunk i is set by the attentional

[4] See [33] for a comprehensive overview of the most recent advancements in the area of cognitive architectures research.

[5] Note that the distinction between 'model' and 'agent' when dealing with cognitive architectures is a blurred one. For clarity's sake we will henceforth use 'agent' to avoid ambiguities with the notion of semantic model (ontology). In general, an agent is a cognitive model that dynamically interacts with the environment.

weight W_k, given the element k and the strength of association S_{ki} between k and the i. The third term states that, under partial matching, ACT-R can retrieve the chunk that matches the retrieval constraints to the greatest degree, combining the similarity Sim_{li} between l and i (a negative score that is assigned to discriminate the 'distance' between two terms) with the scaling mismatch penalty MP. The final factor of the equation adds a random component to the retrieval process by including Gaussian noise to make retrieval probabilistic.

The intertwined connection between declarative and procedural knowledge, weighted by stochastic computations, represents the necessary substrate for realizing at the computational level the functionalities outlined in section *B*: more specifically, we claim that ACT-R can successfully be used to emulate human behavior in selecting and executing defense strategies, matching input data from on-going cyber attacks to deeply structured background knowledge of cyber operations. In the past, ACT-R architecture has been successfully used in context where integrating declarative and procedural knowledge was also a fundamental issue, e.g. air traffic control simulations [24].

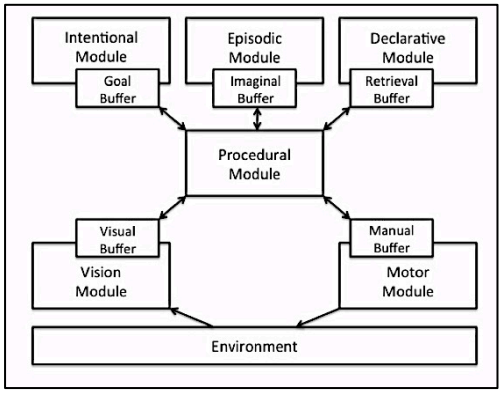


Figure 1 ACT-R Modular Structures

D. Augmenting ACT-R with cyber security ontologies

The development of cyber security ontologies is a critical step in the transformation of cyber security from an art to a science. In 2010, the DOD sponsored a study to examine the theory and practice of cyber security, and evaluate whether there are underlying fundamental principles that would make it possible to adopt a more scientific approach. The study team concluded that:

The most important attributes would be the construction of a common language and a set of basic concepts about which the security community can develop a shared understanding. A common language and agreed-upon experimental protocols will facilitate the testing of hypotheses and validation of concepts [25].

The need for controlled vocabularies, taxonomies, and ontologies to make progress toward a science of cyber security is recognized in [26] and [27] as well. In the domain of cyber security, the ontologies would include, among other things, the classification of cyber attacks, cyber incidents, and malicious and impacted software programs. From our point of view, which seeks to accurately represent the human-side of cyber security, we also expand our analysis to: *(i)* the different roles that system users, defenders and policy makers play in the context of cyber security; *(ii)* the different jobs and functions that the members of cyber defender team play and the knowledge, skills and abilities needed to fulfill these functions. In order to reduce the level of effort, we will reuse existing ontologies when possible [6] and only create new ontologies that support the use cases we select.

The decentralization of knowledge organization and maintenance to a variety of interconnected ontology modules leverages a shared bridging component, i.e. BFO reference ontology[7]: in this sense, BFO plays the role of the common semantic infrastructure to define, populate and update multiple context-driven cyber ontologies. The various modules will be encoded in W3C language OWL[8]: the process of porting them into ACT-R is managed automatically at the architecture level by built-in LISP functions, which are able to *a.* read and interpret the XML-based syntax of the semantic model and *b.* convert it into ACT-R declarative format. A set of broad schemas drives this conversion process: for instance, the direct mapping between the "chunk-type" primitives in ACT-R and classes in the ontologies has been designed. Further schemas at a narrower level of granularity will be provided, as engineered for an analogous framework presented at STIDS 2012 [28].

IV. Cognitive Simulations of Cyber Operations

A. Experimental Design

The first objective of building an intelligent system endowed with adequate representation of cyber security knowledge is to use it in scalable synthetic environments for training human decision-makers. In addition, once the system has incorporated the necessary rational capabilities (defined in the previous section) and learned the dynamics of team interaction, we aim at testing the possibility of deploying it as an autonomous defensive agent in virtual cyber operations. In order to achieve the necessary degree of robustness and dependability, we plan simulations at different levels of complexity, as follows:

BSE — *Basic Synthetic Environment*: two ACT-R agents face each other playing the role of assailant and defender;

HSE — *Hybrid Synthetic Environment*: an ACT-R agent and a human face each other playing the role of assailant and defender;

HSGE — *Hybrid Synthetic Group Environment*: two teams, each constituted by humans and ACT-R agents face each other playing the role of assailant and defender.

In order to run these incremental simulations, we will initially collect an experimental dataset of cyber attacks, to be split into train and test set. In particular, we will focus on spear phishing attacks, as delineated in section II. The datasets will be organized to instantiate classes and properties of the defined modular ontologies. Each level of the cognitive-based simulation will be conceived as a block composed of multiple

[6] For instance, exploiting material from this portal: http://militaryontology.com/cyber-security-ontology.html
[7] http://ontology.buffalo.edu/bfo/
[8] http://www.w3.org/TR/owl-features/

trials[9]. At the **BSE** level, the simulation aims at assessing the soundness of the cognitive mechanisms executed by the agent, serving also as a system debugging and evaluation of experimental settings. In the **HSE**, the agent will have to compete against humans, whose potentially erratic behavior will be exploited by the agent as a primary source of acquisition of cyber warfare strategies and mental representation of the opponent. Finally, in **HSGE** the scenario will get more complex by shifting to a multi-agent framework, where each defending agent will have to learn intra-group cooperation and build mental representation of the opponent as a group (whose members act complementarily and collectively to harm the defending team).

In the delineated experimental phase we plan to expand our previous work on applying cognitive architectures to decision-making in non-zero sum games [29]: cooperative and conflicting phenomena have been comprehensively studied using game theory [30], in which complex social dynamics are narrowed down to relatively simplified frameworks of strategic interaction. Valid models of real-world phenomena can provide better understanding of the underlying socio-cognitive variables that influence strategic interaction: of course these models need be consistent with the structural characteristics of games, and with the actual everyday situations at hand. In this respect, the goal of the planned cognitive simulations is to study decision-making by deploying computational rational agents in cyber attack "gamified" scenarios.

B. Evaluation plans

As recent studies have shown [31], training users to respond to cyber attacks becomes effective only after several iterations. But high time-costs in training can expose socio-technical systems to harmful consequences, with no chance of recovering stolen information or, even worse, of fully restoring the functionalities of the system. Our approach aims to improve cyber defense strategies and speed up the deployment of counter-measures. In particular, we plan to assess the correspondence between the models' simulations and the human behavior in cyber-operations by analyzing human data in decision-making processes. Accordingly, we will apply different analytical methods, such as computing means and standard errors (for decisions), medians and the 1st and 3rd quartiles (for decision times) — similar approaches have been successfully proposed in [32]. We will encode conversion functions in the system to format the outputs as discrete decisions (e.g. "delete spear phishing email", "scan for malware", "reactivate firewall", etc.). Exploiting ACT-R internal clock module, we will also be able to reproduce decision times at human granularity scale, tracking the relevant stages of the rational decision-making process.

V. APPLICATION FRAMEWORK

So far we have discussed the general requirements and described the high-level cognitive structures of an intelligent system for decision support in cyber warfare. However, a product or a solution based on these requirements and architecture will need to address specific problems in the business domain. Furthermore, the end product would likely require integration with other technical components and frameworks. We see an opportunity to apply the concepts described in this paper for the development of an application capable of assessing and reducing information systems vulnerabilities though live, virtual, and constructive (LVC) simulations. Such an application can support a wide range of cyber defense objectives, including: *(i)* analysis of cyber defense strategies and identification of network; vulnerabilities through simulated attacker-defender interaction in BSE – HSE – HSGE scenarios; *(ii)* training for cyber security personnel with a suitable ACT-R agent simulating the attacker against human players; *(iii)* validation and enhancement of the cognitive models developed with an attack counter-attack scenario. To support LVC simulations, the application will need to work with existing distributed modeling and simulation infrastructures, such as the High Level Architecture (HLA)[10] or Testing and Training Enabling Architecture (TENA)[11]. The key integration activities include:

- Identification and creation of reusable 'objects'. A distributed modeling and simulation framework such as TENA encourages objects representing things such as targets and assets to be reused across simulations[12]. In particular, within the intelligent decision support system, we see opportunities at two levels: 1) creation of reusable objects representing attackers and defenders (these objects can be used to simulate behaviors of the actors); 2) creation of reusable objects representing IT Infrastructure components that could be under cyber attacks (these objects model the commands and instructions that can be sent to various components and their responses).

- Integration of reusable objects in to the middleware layer of the modeling and simulation framework. Figure 2 shows a reusable TENA object (representing cyber attackers) plugged into the middleware layer.

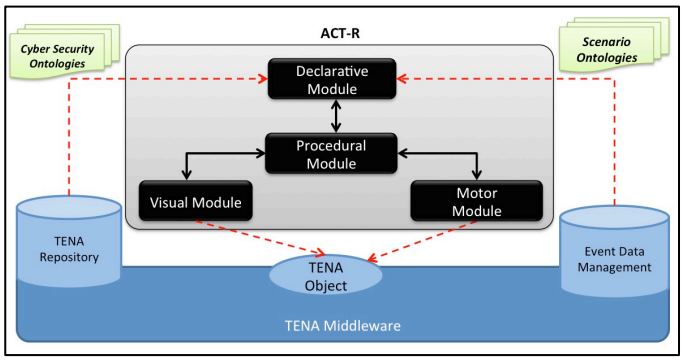
- Implementation of runtime knowledge sharing in the modeling and simulation framework. In the example shown in figure 2, the ACT-R cognitive model (representing the defender) is integrated with knowledge sources incrementally stored in ACT-R declarative memory module: a) modular cyber security ontologies, retrieved from the TENA Repository and; b) the modular ontologies of the scenario [1] , incrementally stored in TENA Event Data Management.

[9] Setting to 100 the number of trials should guarantee a satisfactory level of stochasticity in the results.

[10] http://standards.ieee.org/findstds/standard/1516-2010.html

[11] Test and Training Enabling Architecture (TENA): https://www.tena-sda.org/display/intro/Home

[12] TENA object-oriented modeling features well fit our ontology-driven cognitive system.



[2]

Figure 2 The Cognitive System realized in the TENA framework.

VI. CONCLUSION

The novelty of our approach relies on grounding a decision support system in a broad spectrum of human-level cognitive functionalities blended with highly structured knowledge resources. In particular, by focusing on learning mechanism, context-driven semantic specifications and scalable simulations, the obtained computational system can serve both as a training environment for cyber personnel and as autonomous team member operating in advanced security settings. Our position paper aims at fostering the discussion within the communities of interest and can play the role of a starting platform for a scientific project proposal.

REFERENCES

[1] R. Dipert, "Other-Than-Internet(OTI) Cyberwarfare: Challenges For Ethics, Law, and Policy," *Journal of Military Ethics*, vol. 12, no. 1, pp. 34-53, April 2013.

[2] R. R. Dipert, "The Ethics of Cyberwarfare," *Journal of Military Ethics*, vol. 9, no. 4, pp. 384-410, December 2010.

[3] M.I. Hwang, "Decision Making under time pressure: A model for information systems research," *Information and Management*, vol. 27 , pp. 197-203, 1994.

[4] Symantec. (2013, June) Symantec. [Online]. HYPERLINK "http://www.symantec.com/about/news/release/article.jsp?prid=20130605_01" http://www.symantec.com/about/news/release/article.jsp?prid=20130605_01

[5] Brian Montopoli. (2013, May) CBS News. [Online]. HYPERLINK "http://www.cbsnews.com/8301-201_162-57586624/how-chinese-hackers-steal-u.s-secrets/" http://www.cbsnews.com/8301-201_162-57586624/how-chinese-hackers-steal-u.s-secrets/

[6] (2013, May) Wall Street Journal. [Online]. HYPERLINK "http://online.wsj.com/article/PR-CO-20130530-906764.html?mod=googlenews_wsj" http://online.wsj.com/article/PR-CO-20130530-906764.html?mod=googlenews_wsj

[7] J. C. Forsythe, A. Silva, S. Stevens-Adams, and J. Bradshaw, "Human Dimensions in Cyber Operations Research and Development Priorities," SANDIA Report 2012-9188, Technical 2012.

[8] K. B. De Greene, *Sociotechnical systems: factors in analysis, design, and management*.: Prentice-Hall, 1973.

[9] N. Guarino, E. Bottazzi, R. Ferrario, and G. Sartor, "Open Ontology-Driven Sociotechnical Systems: Transparency as a Key for Business Resiliency," in *Information Systems: Crossroads for Organization, Management, Accounting and Engineering.*, 2012, pp. 535-542.

[10] John R. Anderson and Christian J Lebiere, *The Atomic Components of Thought*.: Erlbaum, 1998.

[11] John R. Anderson and Christian Lebiere, "The Newell Test for a theory of cognition," *Behavioral and Brain Sciences*, vol. 26, no. 5, pp. 587-637, 2003.

[12] Allen Newell, *Unified Theories of Cognition*. Cambridge , Massachusetts: Harvard University Press, 1990.

[13] H. Simon, "Bounded Rationality and Organizational Learning," *Organization Science*, vol. 2, no. 1, pp. 125-134.

[14] M. Wooldridge, *Reasoning about Rational Agents*. Cambridge, MA, United States of America: The MIT Press, 2000.

[15] R. Sun, *Cognition and Multi-agent Interaction*, R. Sun, Ed.: Cambridge University Press, 2006.

[16] Paul Bello, "Cognitive Foundations for a Computational Theory of Mindreading," *Advances in Cognitive Systems*, vol. 1, pp. 59-72, 2012.

[17] A. Oltramari and C. Lebiere, "Knowledge in Action: Integrating Cognitive Architectures and Ontologies," in *New Trends of Research in Ontologies and Lexical Resources*, Alessandro, Vossen, Piek Oltramari, Lu Qin, and Ed. Hovy, Eds.: Springer, pp. 135-154.

[18] J. Ball, S. Rodgers, and K. Gluck, "Integrating ACT-R and Cyc in a large-scale model of language comprehension for use in intelligent agents," in *Papers from the AAAI Workshop*, Menlo Park, CA, pp. 19-25.

[19] B. J. Best, N. Gerhart, and C. Lebiere, "Extracting the Ontological Structure of OpenCyc for Reuse and Portability of Cognitive Models. ," in *Proceedings of the 17th Conference on Behavioral Representation in Modeling and Simulation*, 2010.

[20] S. Douglas, J. Ball, and S. Rodgers, "Large declarative memories in ACT-R," in *Proceedings of the 9th International Conference of Cognitive Modeling*, Manchester, UK.

[21] H. Stuckenschmidt, C. Parent, and S. Spaccapietra, "Modular Ontologies - Concepts, Theories and Techniques for Knowledge Modularization," , 2009.

[22] J. R. Anderson, *How Can the Human Mind Occur in the Physical Universe?* New York: Oxford University Press.

[23] A. Stocco, C. Lebiere, and J. R. Anderson, "Conditional Routing of Information to the Cortex: A Model of the Basal Ganglia's Role in Cognitive Coordination," *Psychological Review*, vol. 117, no. 2, pp.

541-574, 2010.

[24] C. Lebiere,. Constrained Functionality: Application of the ACT-R Cognitive Architecture to the AMBR Modeling Comparison. Mahwah, NJ: Erlbaum, 2005.

[25] The MITRE Corporation, "Science of Cyber-Security," The MITRE Corporation, McLean, VA, Technical 2010 (extract).

[26] D. A. Mundie and D. M. McIntire, "The MAL: A Malware Analysis Lexicon," CERT® Program - Carnegie Mellon University , Technical 2013.

[27] Randall Dipert, "The Essential Features of an Ontology for Cyberwarfare," in *Conflict and Cooperation in Cyberspace - The Challenge to National Security*, Panayotis A Yannakogeorgos and A. B. Lowther, Eds.: Taylor & Francis, 2013, pp. 35-48.

[28] A. Oltramari and C. Lebiere, "Using Ontologies in a Cognitive-Grounded System: Automatic Action Recognition in Video Surveillance," in *Proceedings of STIDS 2012 (7th International Conference on "Semantic Technology for Intelligence, Defense, and Security")*, Fairfax, VA, 2012.

[29] A. Oltramari, C. Lebiere, N. Ben-Asher, and C. Gonzalez, "Strategic Dynamics Under Alternative Information Conditions," in *Proceedings of ICCM 2013 (International Conference of Cognitive Modeling)*, Ottawa, 2013.

[30] A. Rapoport, M. J. Guyer, and D. G. Gordon, *The 2 x 2 game*. Ann Arbor, MI: University of Michigan Press, 1976.

[31] B. M. Bowen, D. Ramaswamy, and S. Stolfo, "Measuring the Human Factor of Cyber Security," *Homeland Security Affairs*, vol. 5, no. 2, 2012.

[32] J. N. Marewski and K. Mehlhorn, "Using the ACT-R Architecture to specify 39 quantitative process models of decision making," *Judgement and Decision Making*, vol. 6, pp. 439-519, August 2011

[33] J.E. Laird, *The SOAR Cognitive Architecture*. USA: The MIT Press, 2012.

[34] C. L. Dancy, F. E. Ritter, and F. E. Berry, "Towards adding a physiological substrate to ACT-R," in *Proceedings of the 21st Conference on Behavior Representation in Modeling and Simulation*, Ameli a Island, FL, 2012, pp. 78-85.

Using a Semantic Approach to Cyber Impact Assessment

Alexandre de Barros Barreto
Instituto Tecnológico de Aeronáutica
São José dos Campos SP Brasil
Email: kabart@ita.br

Paulo Cesar G Costa
George Mason University
Fairfax, VA, USA
Email: pcosta@gmu.edu

Edgar Toshiro Yano
Instituto Tecnológico de Aeronáutica
São José dos Campos SP Brasil
Email: yano@ita.br

***Abstract*—The use of cyberspace as a platform for military operations presents many new research challenges. This paper focuses on the specific problem of assessing the impact of an event in the cyber domain (e.g. a cyber attack) on the missions it supports. The approach involves the use of Cyber-ARGUS, a C2 simulation framework, along with semantic technologies to provide consistent mapping between domains. Relevant information is stored in a semantic knowledge base about the nodes in the cyber domain, and then used to build a Bayesian network to provide impact assessment. The technique is illustrated through the simulation of an air transportation scenario in which the C2 infrastructure is subjected to various cyber attacks, and their associated impact to the operations is assessed.**

***Index Terms*—Impact assessment, cyber-security, Bayesian Networks, C2, semantic technologies.**

I. Introduction

With the increasing automation of processes and systems that are part of critical infrastructures supporting military and civilian operations, the cyber domain became one of most crucial aspects in strategic planning.

As a result, major military players in the world stage started to envision cyberspace as a medium to extend their capabilities, in addition to their existing competencies in the traditional domains (land, air and sea) [1]. However, understanding how cyber operations affect operations and leveraging their effects on the mission are no trivial tasks [2].

To understand the significance of a cyber event in a mission requires mapping physical tasks to their required infrastructure, in a way of creating an integrated view of cyber and physical behaviors. The inherent complexity of this requirement implies, among other things, that an experienced mission analyst must be able to access all relevant data pertaining to the infrastructure and translate it to the support team. Further, this must be done in a way that allows them to understand the real impact of cyber threats not only on the network, but also on the mission it supports.

Many approaches exist to assess cyber impact. However, most are not suitable for supporting complex cyber impact assessment in real situations, where the correlation between kinetic tasks and cyber events needs to be assessed continuously, and with a high temporal resolution. This is a considerable gap that has not been successfully filled, in spite of the relatively large body of research focused on the subject.

This paper presents the Cyber-ARGUS Framework, which leverages semantic technologies to fuse data collected from sensors within the physical and the cyber domains, as well as to retrieve information relevant to the assessment of cyber impact.

The main contribution of Cyber-ARGUS is to provide a mapping of how cyber-events impact tasks in operational level as the mission unfolds. The framework does not create complete maps of vulnerabilities and attacks, or a comprehensive view of how these vulnerabilities and attacks can affect the overall mission. Instead, the framework is meant to provide analysts who need real-time decision support with a simplified situational awareness, which includes understanding what assets are more critical in accomplishing the most important tasks and how these assets are impacted during a cyber attack. As an example from the case study developed for this research, consider the problem of an Air Traffic Security Analyst, who needs to define which elements need to be prioritized to ensure mission success. This analyst must consider data from a large set of different sensors and components, and perform his analysis within very tight time constraints. In his situation, a complete understanding of the current attacks and fault-trees is neither feasible nor necessary, and his task can be accomplished with the simplified mapping and associated impact analysis provided by Cyber-ARGUS.

This paper extends previous work from [3] by addressing how Cyber-ARGUS evaluates the cyber impact on the mission. Among other additions, this paper provides a more detailed explanation on how data from sensors is aggregated, how node-statuses are calculated, and how impact is propagated throughout the network.

To evaluate the Cyber-ARGUS capabilities, we have independently designed a specific air traffic service (ATS) scenario that relies on a new protocol to perform air traffic control in a critical area located at the Campos basin, Brazil. The scenario provides a rich environment to understand how such capabilities can be employed in real life critical operation. The basin, located in the littoral of the Rio de Janeiro state, is a petroleum rich area responsible for 80% of Brazil's petroleum production. ATS missions are critical, happen in real time, and attacks can result not only in considerable economic loss but also in risk of human lives.

This paper is organized as follows. Section II describes the

main concepts of the framework being proposed, as well as a brief survey of the most relevant approaches developed so far to address the problem. Section III conveys a short summary of the Cyber-ARGUS framework, discussing its core ideas. Section IV explains in detail the impact assessment process. Section V presents the study case developed independently for this research, showing the application of Cyber-ARGUS in a specific situation. Section VI presents the results and provides a brief analysis of their significance. Finally, Section VII brings a few considerations and raises issues that must be addressed in future research.

II. Background and Related Research

As implied above, understanding how cyber events affect the missions happening outside the cyber domain is a major requirement for military operations. A common approach for detecting intrusions and system attacks is to use a set of distributed sensors in the network. Preliminary work on this subject focused on specialist or signature-based systems [4], [5].

However, understanding the significance of a cyber-event to a supported mission requires more than identifying attacks and suspect events. It is also necessary to assess their impact on the mission.

Cyber Impact Assessment can be understood as the estimation and prediction of effects on planned or estimated/predicted actions by participants; including interactions between action plans of several players (e.g. Assessing susceptibilities and vulnerabilities to estimated/predicted threat actions given one's own planned actions) [6].

Most approaches attempt to predict how vulnerabilities can be exploited by the enemy (enemy's focus) [7]. Usually, an attack graph [8] that includes vulnerabilities and exploit strategies is generated. Then, an analyst leverages information contained in the graph to calculate impact assessment.

There are a number of issues with this approach. As an example, there are situations in which it is not possible to predict the enemy's behavior, due to the lack of evidence (e.g. on attacks or its detection) resulting in ignorance of self-vulnerabilities or of enemy capabilities. Another issue is the computational problem involved in creating and evaluating the graphs [9].

A recent approach is based on the belief that it is not necessary to identify the enemy's plan or to recognize its actions against one's system. Instead, it is only necessary to know the impact that any plan (ours and enemy's) can have on one's system (mission) [10]. In other words, it is easier to understand the enemy's capabilities and restrictions than it is to predict his behavior. This approach focuses on effects; and does not require one to detect attacks or attackers, but to understand the spectrum of potential effects on the mission. To measure the impact, a model of the mission must be built that includes all critical components that must be identified and monitored. However, [9]–[11] do not describe how to accomplish the mapping between cyber and non-cyber components in detail, as well how to assess the impact of cyber events using real infrastructure data.

An approach to cyber impact assessment was proposed by Holsopple et al. [12]. They define a normalized compromising score, which represents the level of compromise that a node has caused by a specific threat. This method requires defining the threat severity level. One potential approach is to use the Common Vulnerability Scoring System (CVSS). CVSS is a free and open industry standard for assessing the severity of computer system security vulnerabilities.

Even if an analyst knows which attributes are critical to the mission; a second question needs to be answered: how to combine these attributes and generate an index to support coherent and consistent decisions? One strategy is to employ multi-criteria decision making methods (MCDM), a sub-discipline of operations research that explicitly considers multiple criteria in decision-making environments. MCDM provides a set of different approaches that can potentially be used in this cyber-impact assessment. One example is provided in [13], which uses the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) for threat assessment. TOPSIS is a multi-criteria decision analysis method based on the concept that the chosen alternative should have the shortest geometric distance from the positive ideal solution, and the longest geometric distance from the negative ideal solution [14].

Another applicable technique from the MCDM toolbox is presented by [15], which combines Analytic Hierarchy Process (AHP) and TOPSIS for quantifying the degree of security. AHP can be seen as a weight elicitation method based on pairwise comparisons between attributes, and can thus be employed to produce a consistent multi-attribute value structure from experts' input.

Kim and Kang [16] present another MCDM technique to evaluate the critical assets needed to accomplish a mission. Their approach is extremely attractive because it allows for calculating the asset value during a mission using local and global classification. Since the approach involves working in a real-time environment, the authors modified the TOPSIS process to calculate the worst (A-) alternative and the best alternative (A+). Also, a set of maximum and minimum acceptable levels is defined as a means to ensure acceptable performance.

However, this approach has two interrelated limitations. Initially, it was not designed to handle tasks, which are key aspects in defining time sensitive aspects of the mission. As a result of this limitation, the technique becomes less suitable for evaluating distinct phases of a mission. For example, during deployment of a laser-guided bomb by an aircraft, both the soldier illuminating the target (e.g. from a nearby location) as well as the aircraft launching the bomb play equally critical tasks. However, after the ordnance release the aircraft loses its relative importance, since the bomb now relies only on the soldier's laser device in its flight to the target. Such time-sensitive situations cannot be modeled using the approach stated in [16].

In addition to the impact assessment calculations, a key

aspect is to propagate the impact assessed locally in a way of ensuring a coherent understanding of its consequences from a global perspective. A Markov approach approach to model security risk was developed by [17]. However, using Markov processes to propagate impact assessment brings the weakness of the technique's inability to represent non-monotonic dependencies. For instance, in this technique two independent variables must be directly connected by an edge, merely because there are some other variable that depends on both [18].

An alternative for modeling risk propagation is Bayesian Networks (BN). Examples of its use for solving real impact assessment problems can be found in [19], [20]. Li et al. [21] combine CVSS and Attack Graphs in a consistent representation using BNs - which are used to represent the uncertain aspects between exploitation attack paths and the required vulnerabilities. However, we were not able to find a formal description on how to build and elicit the probability tables, which is essential for implementing the technique in a real situation.

Similar the aforementioned work, Singhal and Ou [22] show how to propagate the risk, which is calculated using CVSS metrics, through an enterprise environment using probabilistic attack graph. The latter can be understood as an attack graph that has the associated uncertainty handled by BNs. One problem that is common to all the aforementioned approaches that use BNs to represent uncertainty in attack graphs is that they require complete knowledge of the enemy, a precondition that renders these modeling techniques unrealistic for practical problems.

A different use of BNs is presented by Duan and Babu [23], which periodically collects performance data at three levels: applications, database server, and operating system. The collected data is used to construct probabilistic models for predicting service-level violations. This approach is extremely similar to that of [10], [11], where the impact is calculated by identifying the critical components of mission, their dependences, as well as the effects of their respective failure, and then using a BN to propagate the beliefs to the overall mission.

III. Cyber-ARGUS Framework Review

The goal of this research is to design a framework that enables the understanding of cyber impact within a mission context. This chapter introduces the Cyber-ARGUS framework, which is meant to support this goal. Unlike most approaches cited in Section II, the framework is based on a mission viewpoint approach [10], [11]. From this perspective, the focus is on measuring how the effect generated by a cyber-event intervenes on the results of tasks performed in a mission.

Mapping from the cyber domain to the mission domain requires a few concepts to be defined (e.g. mission, service, and cyber node). The DoD Architectural Framework [24] defines a *mission* as composed by a task (or set of tasks), together with its associated purpose that clearly indicates the action to be taken assigned to an individual or unit. A *service* is a mechanism that enables access to a set of one or more capabilities. In other words, availability of services define which tasks can be performed. The last concept is *cyber node*, which is the element that hosts one or more services.

To understand how an event can produce effects in a mission, Cyber-ARGUS uses an adaptation of the impact dependence graph presented in [7]. The adapted graph includes all relations between tasks; tasks and services; as well as between services and cyber nodes, resulting in a structure that makes it easier to assess the consequences that follow when a node is compromised. Cyber-ARGUS flow of activities is comprised of three main phases [3], [25]: 1) Mission Modeling, 2) Collection Cyber and Mission Situation Awareness, and 3) Cyber Impact Assessment. The first two are treated in parts A and B of this Section, while the latter is explained in more detail in Section IV.

A. Mission Modeling

During the first phase, the core idea is to capture all information about the tasks required to accomplish the mission and consolidate these in an integrated data representation. This allows for a comprehensive analysis to be performed. In our framework, the importance of any given element is measured with respect to its relevance to impact assessment, and includes the associated tasks, the relationships between tasks, objectives, resources required to develop the mission and, finally, the task performer (i.e., entity or set of entities that have the responsibility to execute the mission).

Mission information usually comes from diverse sources, so Cyber-ARGUS ensures consistency of the integrated data representation by means of a mission ontology describing the relevant concepts (tasks, services, nodes, etc.). Semantic technologies also facilitate code reuse, which allow us to avoid having to develop the mission ontology from scratch. Instead, Cyber-ARGUS leverages previous related work by D'Amico et al. [26] and Matheus et al. [27] in its own architecture.

Within this phase of the Cyber-ARGUS activity flow, a mission analyst can design the mission model using any business process language. The goal is to capture the most relevant information of the mission within the model and store it in a semantic Knowledge Base (KB). In the current research, we leveraged previous experience within our group and made the design decision of capturing these aspects using the Business Process Modeling Notation (BPMN) [28]. However, as already mentioned, any business modeling language with the ability to capture the information described above could have been used and, therefore, might be used with the framework in the future.

BPMN was not only convenient as a development tool for the framework, but also proved to be rather suitable for capturing the main aspects of a mission. This is especially true in civilian environments such as air traffic management, nuclear power plants, and others. Its business-oriented notation made it easier to accommodate air traffic domain concepts used in the evaluation part of the research, while also providing a relatively straightforward mapping to the associated concepts in the mission ontology.

The outcome of this first phase includes the mapping of tasks, sequences, and dependencies between them and the required services. Yet, there is no information on where these services are hosted, so the framework queries a service repository and retrieves all information linking IT nodes to their respective hosted services, as well as the network topology depicting the required connectivity.

Once this is accomplished, the framework has all critical information about mission (tasks; service dependencies; and cyber nodes) and can proceed with the next task, vulnerability discovery. The goal now is to locate all vulnerabilities in the infrastructure and store it into the KB to be used in the mission impact assessment phase. This is similar to an infrastructure discovery process, where the framework, using a database, looks for node vulnerabilities that are part of the environment. After this activity, all vulnerabilities and their related impact factors are collected, and Cyber-ARGUS stores this information into the KB. The classification is conducted by nodes, enabling an analyst to perform specific queries relating nodes to vulnerabilities and vice-versa.

The last activity within the Mission Modeling phase to model enemy behavior. Here, the goal is to model known attack-paths using an attack graph. This task requires the existence of a database in which all known attack-paths are described and saved in an appropriate format. To reduce the number of information that Cyber-ARGUS will use during impact assessment phase, we adopted the Cauldron approach developed at GMU [9]. Cauldron uses firewalls and others entrance devices' rules to eliminate implausible scenarios. This strategy reduces the number of nodes and the overall complexity of the original graph, generating a much simpler version that is stored into the Cyber-ARGUS KB as well.

B. Collection Cyber and Mission Situation Awareness

After the Mission Modeling phase, the analyst has a comprehensive view of the mission and the factors that affect its success. That is, the Cyber-ARGUS model is ready to be used; it is now able to collect and correlate infrastructure information, to infer what is pertinent to the mission, and to provide relevant data to calculate cyber impact.

To use this model, the mission analyst needs to collect information from cyber nodes. This will enable him to assess each node's current status, as well as to estimate, during the impact assessment phase, whether the node is able or not to perform the tasks it is expected to perform.

In addition to he node status information, Cyber-ARGUS must collect further data in order to calculate the cyber impact. An example is information about security, which includes attacks events, systems' abuses, etc. This information can be collected from intrusion detection and prevention systems, firewall logs, anti-virus, and other security log system. One important source for this type of data are application and database logs, which can provide a view about how resources are used within the system (e.g., what users logged in, which resource types they used, etc.).

The data collection is one aspect of this phase. The other is the need for correlating and inferring relevant information. To accomplish this, the mission analyst needs to define a set of trigger events (situations), using a language such as the Semantic Web Rule Language (SWRL). SWRL extends a set of OWL axioms to include Horn-like rules, which can be used in conjunction with the OWL knowledge base. The expressiveness achieved by this rule scheme is a key point ensuring the framework's ability to capture aspects that cannot be easily captured using OWL, such as utilization of resources, mission requirements, and others. Furthermore, using the aforementioned rules Cyber-ARGUS can classify from large data sets what elements are relevant, and store it to be used in the next phase, when the cyber impact is assessed.

IV. Cyber Impact Assessment

The cyber impact assessment is defined by four sub-tasks. The first is to generate the Impact Graph, which is a dependence graph [29] that represents mission, as well as the dependence (mission and IT domain) and the influence that each node has on the mission.

The framework will generate three impact graphs, each one representing a security viewpoint (Confidentiality, Integrity, and Availability - CIA). To generate these graphs, the mission analyst needs to inform which tasks he would like to assess and how deep the analysis should be. Using this information, the tasks and assets will be mapped using SPARQL queries [30]. Another key aspect of the framework is its ability to perform plausible reasoning with incomplete data, which enables principled handling of uncertainty. This is achieved by the creation of a Bayesian network (BN) [18] from the impact graph, which we explain later in this Section.

The most critical step in impact assessment is how to measure health node - the ability of the node to provide the services it is responsible for. Our framework measures it through the **operational capacity (OC)**, which is the ability to provide the required resources and services with a certain level of quantity, quality, effectiveness, and cost. In Cyber-ARGUS, this is calculated separately for each of the security views (CIA), enabling the generation of different perspectives.

The OC calculation is presented in Equation 1 below, where $\boldsymbol{OC_x(i)}$ represents the operational capacity of node *i*; $\boldsymbol{sec_x(i)}$ represents its security index, and $\boldsymbol{exp_x(i)}$ represents its exploit index. The security index *x* denotes the security situation of a node for a specific perspective (i.e., confidentiality, integrity, or availability).

$$OC_x(i) = cost \times sec_x(i) \times expl_x(i) \tag{1}$$

Using the same approach of Kim and Kang [16], Cyber-ARGUS uses TOPSIS to aggregate a set of node attributes to define an index. In Cyber-ARGUS, the attributes and the associated weights used to generate the security index are provided by the mission analyst and collected by the event manager.

TOPSIS provides a choice between the shortest geometric distance from the positive ideal solution and the longest

geometric distance from the negative ideal solution. It is crucial because in most network attributes the highest and lowest values convey little or no useful meaning for calculating the security index. An example is the interface's load, in which the highest load value means that interface cannot answer new packets; and the lowest value simply indicates that the interface is not working.

The security index generation starts with creation of a **decision matrix** $(x_{ij})_{m_x n}$, where each of the *m* nodes (i) and their *n* associated attributes (j) are stored. The next step is the normalization of sensor data (Equation 2), which is required for ensuring consistency in additive aggregation techniques. In Cyber-Argus, all attributes are normalized using vector normalization [31], where x_{ij} is the value of the j^{th} attribute of the i^{th} node ($1 \leq i \leq m, 1 \leq j \leq n$).

$$z_{ij} = \frac{x_{ij}}{\sqrt{\sum_{j=1}^{n} x_{ij}^2}} \tag{2}$$

Using normalization matrix, the attributes weights are applied. In the Equation 3, w_j is the weight of the j^{th} attribute.

$$v_{ij} = w_j \times z_{ij} \tag{3}$$

The next step is the calculation of ***zenith*** (A^*) and ***nadir*** (A^-) values, using the equations Equation 4 and Equation 5, where I' is associated with benefit criteria, and I'' is associated with cost criteria [31]. As presented in [16], *max* and *min* values (for performance reasons) are defined by the analyst, based on the maximum and minimum values accepted to accomplish target mission.

$$A^* = v_1^*, ..., v_n^* = (max_j v_{ij} | i \in I'), (min_j v_{ij} | i \in I'') \tag{4}$$

$$A^- = v_1^-, ..., v_n^- = (min_j v_{ij} | i \in I'), (max_j v_{ij} | i \in I'') \tag{5}$$

In the sequence, the Euclidean distances are calculated using Equations 6 and 7.

$$D_j^+ = \sqrt{(\sum_{i=1}^{m} (v_{ij} - v_i^*)^2)}, j = 1, ..., n \tag{6}$$

$$D_j^- = \sqrt{\sum_{i=1}^{m} (v_{ij} - v_i^-)^2)}, j = 1, ..., n \tag{7}$$

Finally, the last step is the calculation of relative closeness to ideal solution (T_j^*). In our framework, this metric represents the security index of a node, $sec_x(i)$, and is calculated using Equation 8. An alternative w is better than y, when $T_w^* > T_y^*$.

$$sec_x(j) = T_j^* = \frac{D_j^-}{D_j^- + D_j^+} \tag{8}$$

The second component of OC is the **exploit index**, $expl(i)$. To calculate it, Cyber-ARGUS retrieves all security information from KB (vulnerability and exploit paths), and verifies the existence of active path attacks to the stored node's vulnerabilities. To compute the index, the possible exploit vulnerabilities are considered via their respective **vulnerability impact factor (V)**, as presented in Equation 9.

$$expl(i) = [\prod_{k=0}^{n} (1 - V_{[k]}(i))] \tag{9}$$

In Equation 9, *i* represents the cyber-node and *n*, the number of vulnerabilities that have a known exploit path that can be explored. This index has the same principles of metrics defined in [32], where the more high score vulnerabilities a node has, the smaller its OC will be and, consequently, the worst will be its ability to provide contracted services reliably.

OC's definition is an essential step in Cyber-ARGUS, as it reflects the model's beliefs. That is, a higher OC means a higher likelihood of accomplishing the mission's goals. The propagation of these beliefs is performed using a BN. In our model, cyber-asset is a deterministic rank node and its values are based on the calculated OC. To simplify the composition of a BN, the OCs will be discretized in three parametric states: high, medium, and low operational capacity. Belief on the reliability of services and tasks are also represented as probabilistic nodes, which states are: unreliable, medium reliability, and reliable. The range of each one of aforementioned states is calculated as defined in [33].

The values of cyber-nodes (i.e. their state variables) are used to assess the beliefs on the reliability of service and tasks. A main issue is how to generate the conditional probability tables (CPT) for the service and task nodes, since it requires time-consuming work from analysts [33]. For example, considering a node that has five parent nodes and each node has two different states, its associated CPT will have 63 values to be elicited (2^5-1, since the last value can be calculated). Cyber-ARGUS addresses this issue by using an automated approach to generate CPTs, as defined in Fenton and Neil [33]. A TNORMAL distribution is used to define the weighted rank node functions, and to calculate the CPTs. Equation 10 illustrates this approach, where **X** is the target variable and **Y** is the conditional evidence.

$$p(X|Y) = \left[FUNC, \quad \frac{1}{\sum_{i=1}^{n}(w_i)}, \quad 0, \quad 1\right] \tag{10}$$

A TNORMAL is similar to a NORMAL distribution, but with its values enclosed within a finite range. In the aforementioned equation, the first parameter is the mean of distribution, which is calculated using **WMIN** (Equation 11) and **WMAX** (Equation 12). The second parameter is the variance, which is calculated using the weight of influence that each parent-node has over the target variable. The last two parameters (values 0 and 1) are the boundary defined for $p(X|Y)$.

$$WMIN = min_{\forall_{i=1,...,n}} \left[\frac{w_i X_i + \sum_{i \neq j}(X_j)}{w_i + (n-1)}\right] \tag{11}$$

$$WMIN = max_{\forall_{i=1,...,n}} \left[\frac{w_i X_i + \sum_{i \neq j}(X_j)}{w_i + (n-1)}\right] \tag{12}$$

In Cyber-ARGUS, weights can be collected during Mission Modeling, using service-level information from the mission analyst. However, they can also be set manually by the analyst, so to reflect his level of uncertain about the fact. In general, the network weight is proportionally inverse to node's distance. For example, if node A hosts a service, its **weight** (w_k) is set to 1 (one). However, if node B is a neighbor of node A and does not host the target service, the framework applies Equation 13, where *r* is the distance of hosted node.

$$w_k = \frac{1}{r} \tag{13}$$

Further, when a dependent node (service or task) connects parent nodes using **OR** relationship, the **WMAX** function is used. Conversely, if it has an **AND** relationship, the framework uses the **WMIN** function.

The cyber impact on the mission is calculated after the belief propagation process, which occurs step-by-step from cyber-assets to services and from services to tasks. A more formal representation of impact on the mission beliefs, $imp(x)$, is presented in Equation 14, where its values are calculated from a joint probability distribution. In the equation, *X* is the mission result node and *Y* is the set of parents of this node.

$$imp(X) = p(X|Y) = p(Y|X) \times \prod_{i=1}^{n} p(X_i) \tag{14}$$

V. Study Case - Air Traffic Scenario

To evaluate the framework, we have independently developed an air traffic scenario representing the Air Traffic Control operations in the Campos Basin. This is a petroleum rich area in the Rio de Janeiro state that is responsible for 80% of Brazil's petroleum production, which is prospected and explored from oceanic fields. The operation relies on heavy helicopter traffic between the continent and oceanic fields during daytime, with an average of 50 minutes per flight.

To support this operation, Brazil has an Air Control Center (ACC) in Macaé (Rio de Janeiro). This center has a radar station that supports the surveillance service within the terminal. However, the oil platforms are located at sites that are more than 60NM from Macaé. Helicopter flights are carried out at low altitude, so there is no radar coverage close to the oil platforms and thus the Air Traffic Service (ATS) has to be based on non-radar procedures. This significantly reduces efficiency of air operations.

The Brazilian Government solution currently under study includes adopting the Automatic Dependent Surveillance-Broadcast (ADS-B) technology. The strategy is to supplement radar coverage in the oceanic air space. The ADS-B operation is based on using radios to transmit and receive aircraft position information generated through the satellite GNSS GPS via a data link. The radios work as relay agents, sending positional information to a central node. This data is then integrated to an ADS-B Server, which supports air traffic controllers in managing the air traffic.

This new technology has a set of security issues. A complete survey of ADS-B's vulnerabilities, different ways to exploit it, and the importance in protecting it is presented in [34].

Due to its criticality and vulnerability, the Campos Basin's scenario is a good candidate to validate the Cyber-ARGUS framework. The scenario was implemented using a complex, distributed simulation/emulation environment, the C2 Collaborative Research Testbed [25].

The C2 Collaborative Research Testbed scenario includes all ADS-B radio-stations existing in the area, a set of simulated helicopters. It provides a realistic environment, suitable for evaluating all phases of the Cyber-ARGUS framework. In the experiments, Cyber-ARGUS was used to build the Impact Dependence Graph, which has all tasks, services and nodes required to asses the cyber-impact on the typical mission with that scenario. As an example, to accomplish goal "M1" it is required to perform tasks "Manage Traffic" and "Deconflict Traffic," which were part of the experiments. The resulting graph was used to build the BN, and the services and tasks beliefs were calculated using WMIN and WMAX function, enabling that impact on the mission can be calculated. The preliminary results of these experiments are discussed in Section VI below.

VI. Preliminary Results and Discussion

In the Cyber-ARGUS evaluation experiments, each round consumed approximately two hours. During this time, a set of attributes of the cyber nodes were collected and their associated OCs were calculated. The OCs were then used to feed the BN and calculate the impact.

In this initial evaluation, the focus was in measuring the availability attributes in response to a campaign of Deny-of-Service attack (DoS). A DoS is an attempt to make a machine or network resource unavailable to its intended users. This attack aims to interrupt the service that is required to be performed for achieving a given mission task. The campaign was performed during three times, and in each iteration the required values were collected and the final impact assessed using the full Cyber-ARGUS process. In the first attack, the target included the ADS-B radios P20 and MAC. These two radios are important to the mission because they cover most of the oil platforms. When they fail, some platforms lose their ADS-B coverage, which results in the ATC reverting back to a lesser operation mode with increased separation between aircraft. The second attack aimed to deny all network, and all radio's nodes and servers were attacked. In the last campaign, the attack was specifically against the ATC-SIM. This is a server responsible for processing all tracks, fusing them and displaying on the ATC's visualization. It provides all information needed for the controllers' situational awareness.

The results of the first and second attacks are shown in the Figure 1. In the graphic, the beliefs for nodes OC, service and goal are represented. All values were normalized, and the most important information is the trend of attributes. Note that Mission Goal (M1) is completely insensitive to variations in the P15 radio. However, attacks on nodes MAC and P20

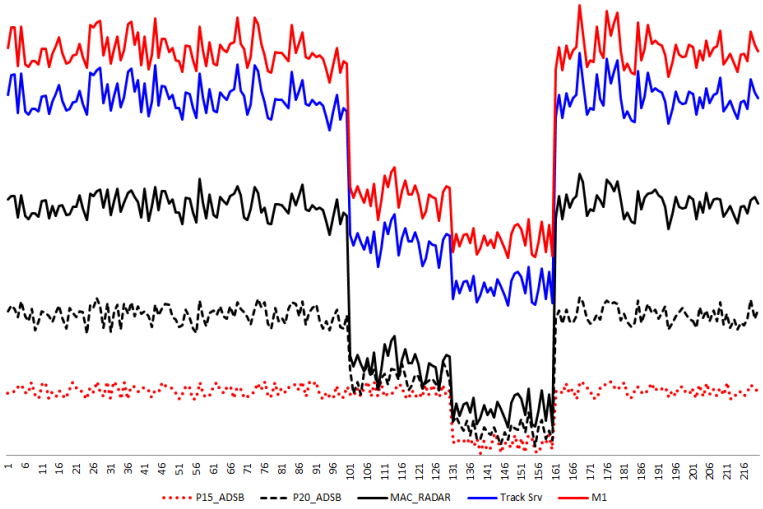


Fig. 1. Attack on P20 and MAC

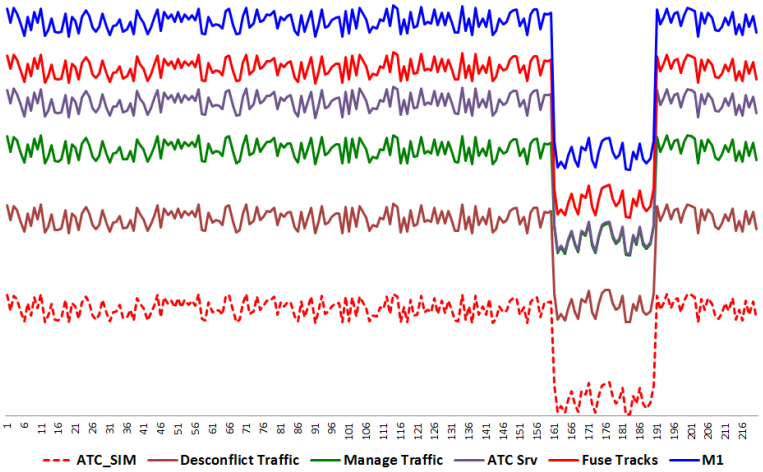


Fig. 2. Attack on ATC Server

(between 100 and 150 slot-time) resulted in a decrease in the track service and the goal beliefs. This shows that OC is a good estimator of mission assurance [11].

The last attack was more critical, as it happens on the main server that supports the mission. The results clearly show that all process automation was denied, decreasing the belief that mission can be performed with the same level of success than in a normal situation. Figure 2 shows that when the server is down, controllers revert back to conventional operation. This results in a great decrease of operational performance, although the mission still continues to happen. As in the early example, during the ATC attack the trend line is the same to the server, to the services it hosts, and to the mission goal.

VII. Final Remarks

Cyber-ARGUS is a framework that enables the calculation of the impact that actions within the cyber domain have over elements in the operational domain. This allows for a large spectrum of analysis on complex Command and Control operations (Military, Civil, and others), where events that happen in one dimension will be reflected in other dimensions. The framework also enables a better understanding of the critical events that affect the environment and have impact on the mission. This capability can also be used to develop more accurate defense/offensive plans and scenarios in critical applications.

In this paper, we showed the use of a knowledge base to generate the impact graph, which is then used to propagate

the nodes effects beliefs to services and tasks.

This is a research in progress in an area where clear answers are usually not attainable, mostly due to the complexity but also to the subjectivity involved in assessing impact in an ongoing operation. Currently the framework is being extended to provide new capabilities and allow its use in increasingly richer and more complex scenarios. One of the limitations of the current implementation is its inability to change the network topology and reflect the effect inside the BN, which is an important aspect given the constant network changes due to sensor reallocation, losses, and similar phenomena. Another limitation is the lack of a proper modeling of the enemy behavior (attack graph), which is needed to calculate the exploit index, and generate accurate information to represent the OC index. Finally, it's necessary more complex and different scenarios, providing confidence to apply method in general Command and Control scenarios.

ACKNOWLEDGMENTS

The authors would like to thank VT MÄK for providing all tools and support to develop the Testbed. They would also express their gratitude to the anonymous reviewers for their careful work and insightful comments.

REFERENCES

[1] M. G. W. T. Lord, "Cyberspace operations: Air force space command takes the lead," *High Frontier - The Journal for Space & Missile Professionals*, vol. 5, pp. 3–5, 2009.

[2] V. N. E. Brown, "Difficulties encountered as we evolve the cyber landscape for the military," *High Frontier - The Journal for Space & Missile Professionals*, vol. 5, pp. 6–8, 2009.

[3] A. B. Barreto, P. Costa, and E. Yano, "A semantic approach to evaluate the impact of cyber actions to the physical domain," in *Semantic Technologies for Intelligence, Defense, and Security 2012.*, P. C. G. Costa and K. B. Laskey, Eds., vol. 966, no. ISSN 1613-0073. CEUR-WS, October 2012, pp. 64–71. [Online]. Available: http://ceur-ws.org/Vol-966/STIDS2012_T08_BarretoEtAl_EvaluateImpactOfCyberActions.pdf

[4] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, vol. 13, pp. 222–232, 1987.

[5] T. Bass, "Multisensor data fusion for next generation distributed intrusion detection systems," in *IRIS National Symposion*, 1999.

[6] É. Bossé, J. Roy, and S. Wark, *Concepts, models, and tools for information fusion*. Artech House ˆ eBoston Boston, 2007.

[7] G. Jakobson, "Mission cyber security situation assessment using impact dependency graphs," in *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, 2011, pp. 1–8.

[8] B. Schneier, "Attack trees: Modeling security threats," Dr. Dobb's journal, December 1999.

[9] S. Jajodia, S. Noel, P. Kalapa, M. Albanese, and J. Williams, "Cauldron mission-centric cyber situational awareness with defense in depth," in *MILITARY COMMUNICATIONS CONFERENCE, 2011 - MILCOM 2011*, 2011, pp. 1339–1344.

[10] S. Musman, M. Tanner, A. Temin, E. Elsaesser, and L. Loren, "Computing the impact of cyber attacks on complex missions." in *2011 IEEE International Systems Conference (SysCon)*, 2011, pp. 46–51.

[11] S. Musman, M. Tanner, A. Temin, E. Elsaesser, and L. Loren, "A systems engineering approach for crown jewels estimation and mission assurance decision making." in *IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, 2011.

[12] J. Holsopple, S. J. Yang, and M. Sudit, "Tandi: threat assessment of network data and information," in *Proceedings of SPIE, Defense and Security Symposium*, vol. 6242, April 2006, pp. 114–129. [Online]. Available: http://dx.doi.org/10.1117/12.665288

[13] Q. Changwen and H. You, "A method of threat assessment using multiple attribute decision making," in *Signal Processing, 2002 6th International Conference on*, vol. 2, 2002, pp. 1091–1095 vol.2.

[14] C. L. Hwang, *Multiple Attribute Decision Making: Methods and Applications*, ser. Lecture Notes in Economics & Mathematical Systems. Springer-Verlag, 1981.

[15] N. Liu, J. Zhang, H. Zhang, and W. Liu, "Security assessment for communication networks of power control systems using attack graph and mcdm," *Power Delivery, IEEE Transactions on*, vol. 25, no. 3, pp. 1492–1500, 2010.

[16] A. Kim and M. H. Kang, "Determining asset criticality for cyber defense," ONR, Memorandum Report 55-6334, 2011.

[17] Y.-G. Kim, D. Jeong, S.-H. Park, J. Lim, and D.-K. Baik, *Modeling and Simulation for Security Risk Propagation in Critical Information Systems*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, vol. 4456, ch. Computational Intelligence and Security, pp. 858–868.

[18] J. Pearl, "Markov and bayes networks: A comparison of two grapgraph representations of probabilistic knowledge," University pf california, Tech. Rep., 1986.

[19] J. Wu, L. Yin, and Y. Guo, "Cyber attacks prediction model based on bayesian network," in *Parallel and Distributed Systems (ICPADS), 2012 IEEE 18th International Conference on*, 2012, pp. 730–731.

[20] S. van Gosliga, R. van Katwijk, and P. van Koningsbruggen, "Real-time traffic monitoring with bayesian belief networks," in *11th World Congress on Intelligent Transport Systems (ITS-2005)*, 2005.

[21] J. Li, X. Ou, and R. Rajagopalan, "Uncertainty and risk management in cyber situational awareness," in *Cyber Situational Awareness*, ser. Advances in Information Security, S. Jajodia, P. Liu, V. Swarup, and C. Wang, Eds. Springer US, 2010, vol. 46, pp. 51–68. [Online]. Available: http://dx.doi.org/10.1007/978-1-4419-0140-8_4

[22] A. Singhal and X. Ou, "Security risk analysis of enterprise networks using probabilistic attack graphs," National Institute of Standards and Technology, Tech. Rep., 2001.

[23] S. Duan and S. Babu, "Proactive identification of performance problems," in *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, ser. SIGMOD '06. New York, NY, USA: ACM, 2006, pp. 766–768. [Online]. Available: http://doi.acm.org/10.1145/1142473.1142582

[24] DoD, *DODAF. DoD Architecture Framework Version 2.0 - Volume 1: Introduction, Overview, and Concepts.*, DoD Std., 2009.

[25] A. B. Barreto, M. Hieb, and E. Yano, "Developing a complex simulation environment for evaluating cyber attacks," in *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2012.*, vol. 12248, December 2012, pp. 1–9.

[26] A. DAmico, L. Buchanan, J. Goodall, and P. Walczak, "Mission impact of cyber events: Scenarios and ontology to express the relationships between cyber assets, missions, and users." AFRL/RIEF, Tech. Rep. OMB No. 0704-0188, December 2009.

[27] C. J. Matheus, M. M. Kokar, K. Baclawski, J. A. Letkowski, C. Call, M. Hinman, J. Salerno, and D. Boulware, "SAWA: An assistant for higher-level fusion and situation awareness," *Proceedings of SPIE*, vol. 5813, no. 1, pp. 75–85, 2006. [Online]. Available: http://link.aip.org/link/?PSI/5813/75/1\&Agg=doi

[28] OMG, *Business Process Model and Notation (BPMN) 2.0*, http://www.omg.org/spec/BPMN/2.0, OMG Std., 2011.

[29] F. Balmas, "Displaying dependence graphs: a hierarchical approach," in *Proceedings of the Eighth Working Conference on Reverse Engineering (WCRE'01)*, ser. WCRE '01. Washington, DC, USA: IEEE Computer Society, 2001, pp. 261–. [Online]. Available: http://dl.acm.org/citation.cfm?id=832308.837144

[30] E. Prud and A. Seaborne, *SPARQL 1.1 Overview*, W3C Std., 2008. [Online]. Available: http://www.w3.org/TR/rdf-sparql-query/

[31] K. Yoon and C. Hwang, *Multiple Attribute Decision Making An Introduction*. SAGE, 1995.

[32] B. J. Argauer and S. J. Yang, "Vtac: virtual terrain assisted impact assessment for cyber attacks," in *Proc. SPIE 6973, Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security*, B. V. Dasarathy, Ed., vol. 6973, 2008.

[33] N. Fenton and M. Neil, *Risk Assessment and Decision Analysis with Bayesian Network*. CRC Press, 2013.

[34] D. McCallie, J. Butts, and R. Mills, "Security analysis of the ads-b implementation in the next generation air transportation system." *International Journal of Critical Infrastructure Protection*, vol. 4, pp. 78–87, 2011.

Analyzing Military Intelligence Using Interactive Semantic Queries

Rod Moten
Data Fusion and Analytics
Sotera Defense Solutions
Aberdeen Proving Grounds, MD, USA
rod.moten@soteradefense.com

Abstract— **We describe a strategy for performing semantic searches for analyzing military intelligence. Our strategy allows the analyst and the query engine to work together to reduce a complex query into simpler queries. The answers for the simpler queries are combined into answers for the original query. The queries can be refined using rules defined by the analyst or analytics created by a data scientist. Our strategy uses an alternative approach to semantic modeling than the state-of-the-art approaches based on OWL. OWL is an implementation of a branch of mathematical logics designed specifically for semantic modeling called description logics. Our strategy uses a branch of mathematical logics called type theory. We use type theory because of the long history of developing systems based on type theory for reasoning interactively. We demonstrate with an example how the strategy can be used to answer questions posed by analysts that couldn't be answered using conventional methods.**

Keywords: semantic search; military intelligence; analytics; type theory; ontology; semantic modeling; interactive theorem proving

I. INTRODUCTION

"The Army is working closely with the intelligence community and other Defense Department partners, including the Navy, in developing cloud-based systems for battlefield intelligence."[1] The goal of the U.S. Army is to fulfill theater intelligence requirements using these systems as much as possible [2]. For example, suppose an analyst created a hypothesis that a family within an Afghan village is responsible for several IEDs. The analyst may use the Cloud to determine which families have connections to hostile organizations. The data may be in the Cloud that *directly links* a family to a hostile organization. For example, suppose the Sadat Baba family [2] is a member of the village and intelligence data contains the triple (Sadat Baba shares-profit Taliban). In the triple, Sadat Baba is the subject, shares-profit is the predicate and Taliban is the object. On the other hand, the intelligence data may only contain data that *indirectly links* the family to a hostile organization. These links have to be inferred either deductively or inductively from the data. For example, it may be possible to infer that Sadat Baba and the Taliban have common interests because both Sadat Baba and the Taliban attacked the Dalazak family. This could be inferred by applying the following rule. If a family and an external organization attack another family in the same tribe or village, then the external organization and the attacking family have common interests. Or it could be inferred using network analysis because Sadat Baba and Taliban both are linked to Dalazak by the same relationship within the same subgraph.

The current state-of-the-art for military intelligence analysis focuses on the analyst using visual aids and various retrieval techniques, such as faceted search and querying, to perform this inference manually [3]. Military intelligence analysis systems should focus on developing strategies for reducing the manual work performed by analyst by incorporating more automated methods. The effort for searching for data can be reduced if the query engine automated some of the work the analysts performed manually. This means the query engine would need to have analytics that automate some of the inductive and deductive reasoning performed by the analyst.

Some of the analytics used in a query or search may be different from the analytics used in ETL (extraction, transformation, and load). In ETL, analytics are used primary for entity and feature extraction, entity resolution, and entity fusion [4]. In this case, the analytics aren't used to answer a query posed by an analyst. The analytics we are interested in, such as multi-relational link predication [5] [6], occur after ETL. The analytics will be performed after the data has been mapped into a graph-like structure, such as RDF or DRIF [7], [8]. Therefore, the query engine will need the ability to express the behavior of the analytic in terms of the underlying semantic network.

The behavior of an analytic can be specified as the logical implication of the postcondition from the precondition. The precondition is a logical statement that must be satisfied in order for the analytic to produce valid output. The postcondition is a logical statement that describes the characteristics of the concepts and relationships produced by the analytic. In the simplest case, the specification of an analytic could be defined as $\phi \rightarrow \psi$, where ϕ is the precondition and ψ is the postcondition. For example, the precondition of the analytic for associating a family and an external organization would be 'for any $?f \in$ Family and $?n \in$ National Organization there exists a $?g \in$ Family such that $(?f \text{ attack } ?g)$ and $(?n \text{ attack } ?g)$'. The postcondition would be '$(?f \text{ common-interests } ?n)$'.

The examples above may mislead the reader to believe that first-order logic would be sufficient for expressing the precondition and postcondition. However, first-order logic doesn't support cases when analytics can operate on a set of concepts and roles. For example, consider an analytic that uses numeric calculations to determine relationships, such as common neighbor algorithms [9]. Such an analytic only cares about relationships or edges between nodes. So it can operate on any concept. For example, if the analytic uses a common neighbors algorithm for determining the common-interests relationship, then the precondition would be 'for any $?f \in ?F$ and $?n \in ?N$ there exists a $?g \in ?F$ such that $(?f\ ?A\ ?g)$ and $(?n\ ?A\ ?g)$'. In this precondition, we parameterized the concepts for the attackers, Family and National Organization, and we parameterized the attack relationship. We can also parameterize the postcondition. For example, a data scientist may be able to improve the analytic to infer a stronger relationship between the attackers using additional information about them. In this case, the postcondition would be 'there exists a $?Q \subseteq \text{common-interests}$ such that $(?f\ ?Q\ ?n)$'.

The query engine could use the specification of an analytic as a rule for solving an intelligence requirement. If the specification of an analytic is $\phi \rightarrow \psi$ and the intelligence requirement can be reduced to ϕ, then the solution to the intelligence requirement would be the concepts and relationships that the analytic produces that satisfy ψ.

In this paper, we present a strategy for analyzing intelligence data using an *interactive query language*. When a user specifies a query, the query engine solves the query by refining it into new queries. If any of the new queries cannot be answered, it asks the user to assist it. The user assists the query engine by specifying an analytic or rule that can solve the query or reduce the query into new queries. This process continues until all queries are answered or until there is a query that cannot be answered. The query engine keeps track of this process and combines the answers from the generated queries into an answer for the original query. At the heart of the query language is the type theory TT-IQ, *Type Theory for Interactive Querying*. The query engine for TT-IQ consists of a framework that allows a data scientist to define analytics that can be included in query processing and for analysts to add new rules.

This paper is outlined as follows. First, we present work related to our strategy. Then, we use an example to demonstrate interactive querying. Next we give an overview of TT-IQ. Finally, we conclude the paper with a discussion of our strategy.

II. Related Work

Our strategy is similar to approaches that use a semantic network or ontology for refining queries. These approaches, such as QUICK [10], LISQL [11], and query rewriting [12] use semantic information to enhance a query supplied by a user. These approaches use a semantic network and stepwise refinement to create semantic queries. Our approach, on the other hand, uses stepwise refinement for query execution.

Researchers at GMU have spent over 15 years developing strategies that could be used for interactive querying [13]–[17]. Their approaches use heuristics to perform inductive and deductive reasoning. They also use machine learning to find new rules to add to the knowledge base. Our strategy support inductive and deductive reasoning except we use proof-theoretic methods used in interactive theorem provers. Theorem provers, such as NuPrl [18], Coq [19], and Isabelle [20], use interactive methods for developing formal mathematical proofs. In these systems, the assertions are *type judgments*. Type judgments are logical statements that ask which objects belong to a specific type. The types can be defined to resemble logical statements, such as $P \wedge Q \rightarrow R$. We use this same technique in our query language. However, our type theory differs from the state-of-the-art in order to support semantic modeling.

K-DTT [21] and S-DTT [22] are type theories that use an *extensional* approach to semantic modeling. Description logics also use an extensional approach to semantic modeling. This means that A-Box statements or an external source, such as a database, has to be used to determine that an object belongs to a concept. In TT-IQ, concept membership is *intensional*. In other words, we determine whether an object belongs to a concept based on how the object was constructed. Therefore there is no need for A-Box statements or external sources to determine concept membership.

Type theory isn't the only method of using higher-order logic for semantic modeling. Classical higher-order logics have been used for semantic modeling as well [23]–[26]. However, the query languages used in these approaches do not allow a user to define analytics to be incorporated into the query engine.

Rule-based approaches for semantic search, such as Tuple-Generating Dependencies [27], have the capability to incorporate rules specified by a user into the query answering process. However, these approaches are implemented using first-order reasoning techniques from logic programming. As a result, they will not be able to support domain metamodeling. Our strategy, on the other hand, will support finding concepts and relationships that meet specific criteria.

III. Interactive Querying

In this section, we give an overview of how interactive querying is performed. Interactive querying is analogous to proving a type judgment. Informally, a typing judgment consists of a goal and a set of assumptions. The goal is the assertion we want to prove. The assumptions represent facts and statements from the knowledge base. Therefore, it consists of logical statements about the semantic meaning of concepts and relationships.

Fig. 1 shows an example proof created from an interactive query for attacks in a region that may have involved a specific person. We assume the analyst only knows that the person has a set of features observed by an interrogator, such as facial marks, height, and the number of a cell phone belonging to the detainee. This query is stated as a judgment that appears as the root of the proof tree. Here we state the judgments informally in natural language. The assumptions in the judgments in Fig. 1 contain a definition of a type representing the concepts Attack and Person; an object representing the features of the detainee;

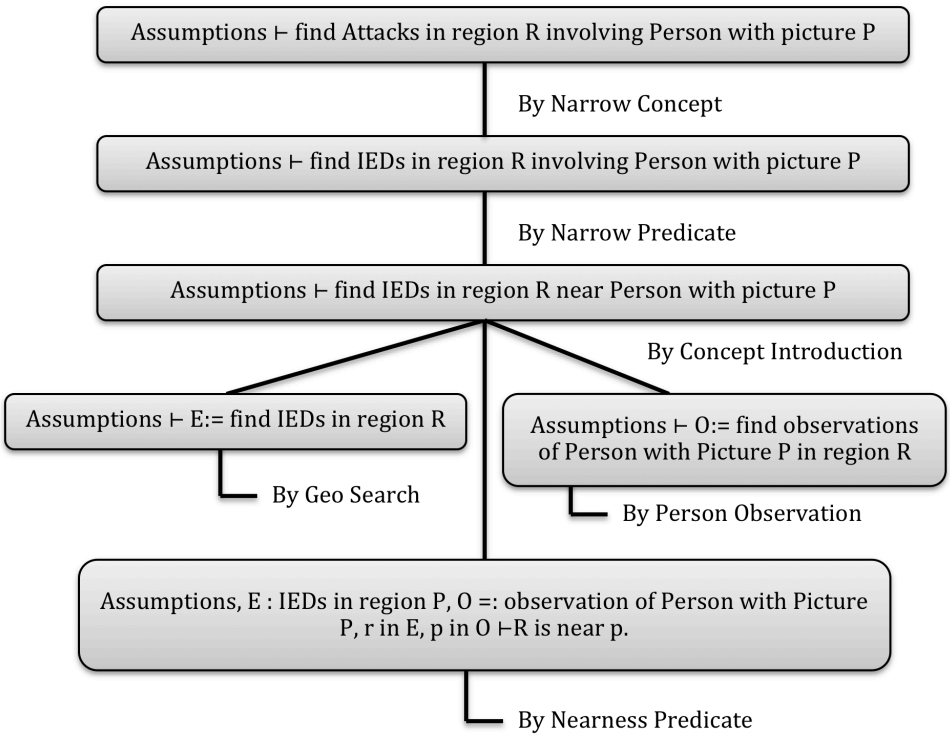


Fig. 1. Example proof created from an interactive query

features of the detainee; declarations of the predicate symbols, such as involving; and definitions of data types that represent cell phone numbers and regions. The assumptions also contain a taxonomy of properties and features. The taxonomy is defined as a partial ordering on predicate symbols and attribute names.

The query engine determines which *tactics* can be applied based on whether the conclusion of a tactic matches the goal of a judgment. The query engine uses the *type compatibility relationship* [28] of TT-IQ to perform matching. When two types T and T' are compatible, it means that an object in T can be converted to an object in T' and vice versa. Type compatibility is an extension of the subtyping relationship in TT-IQ. We give an overview of subtyping in "Overview of TT-IQ".

Each tactic has a conclusion and zero or more antecedents. When a tactic matches a judgment, the query engine creates new judgments for each antecedent. For example, the Narrow Concept tactic is a built-in tactic that replaces a type T in a term with a subtype of T. This tactic has two antecedents. One antecedent is a judgment that asserts the replacement type is a subtype of the substituted type. The other antecedent is the same as the conclusion except all occurrences of the supertype are replaced with the subtype. Fig. 1 only shows the second antecedent because the first antecedent can be proved automatically using TT-IQ's subtyping relationship.

In practice, the query engine will only show the judgments that require assistance from the user. There may be multiple tactics that can be applied to a judgment. The compatibility relationship can rank the tactics that match best. However, if two tactics have the same rank, then the user will need to select the tactic to apply.

Narrow Predicate is similar to Narrow Concept, except Narrow Predicate replaces a predicate symbol with another predicate symbol. Therefore, Narrow Predicate has antecedents to prove that the replacement predicate is a *sub-property* of substituted predicate. This requires two antecedents: an antecedent to prove that the replacement property is a sub-property of the existing property and an antecedent to prove that the type of the sub-property is a subtype of the type of the super-property. The first antecedent has to be proved using the taxonomy of the properties and features. The tactic also has a third antecedent that contains the sub-property instead of the super-property. The proof tree in Fig. 1 shows the judgment produced from the third antecedent. In particular, it shows "involving" replaced with "near".

The judgments produced by Concept Introduction illustrate the need for *dependent judgments*. Normally in type theory and in sequent calculus, judgments of the same parent are independent of each other. However, when using interactive proofs to query for data, terms created on one branch could be used in a judgment on a different branch. Notice that two of the goals produced by Concept Introduction contain $:=$. This special constructor informs the query engine to create a reference to the term that satisfies the type on the right hand side of $:=$. At some point, a tactic will be invoked that uses an analytic to create objects or find objects in the knowledge base to bind to the reference.

The Geo Search tactic uses an analytic to bind a reference to a collection of terms that are within a specific region. Geo

Search uses the goal as the criteria to search for objects within a specific region in the knowledge base. More specifically, Geo Search has a conclusion that has a type that contains one attribute, location. The type of location is the supertype of all types that could be used as the criteria for a geospatial search, such as KML. This type matches any type that has an attribute that is compatible with KML and the name of the attribute is interchangeable with location. The taxonomy determines which attributes are interchangeable with location. The tactic uses this attribute as the search criteria. In practice, the tactic may also require a time range.

Person Observation is a tactic that is created by a data scientist. In other words, it is an analytic that creates concepts inductively. This means it creates the definition of a type by generalizing existing data. Let's assume that Person Observation examines SIGINT data for calls originating form or made to the telephone number of the cellphone in the possession of a detainee. The time and location of the each call is used as an *observation point* of the person. In theory, a new triple is added to the knowledge base that links the person to the time and location of the call. In practice, the query engine may not create the triples. Instead, the query engine may define a way to create the triples on demand without altering the knowledge base unless directly instructed to do so. This approach is essential for cloud-scale data because it doesn't perform any destructive modifications if the analyst wants to back out changes. Instead the query engine could have a local cache containing the new triples. The modifications could be made permanent only when explicitly specified by the analyst or a data scientist.

The antecedent and conclusion of a tactic based on an analytic is determined by the precondition and postcondition of the analytic. The system uses the precondition as the antecedent and the postcondition as the conclusion. For Person Observation the postcondition states that there exists a concept P where each object is a Person that has the features of the detainee and whose locations correspond to the locations of the cell phone. The precondition requires that there exists a person in the knowledge base who has the features of the detainee. To satisfy the precondition, the analyst will need to add the detainee to the knowledge base. The precondition also requires that we can determine the locations of the cell phone. We assume another analytic will determine this location. For brevity, assume that the query engine can prove this automatically. As a result, it isn't included in Fig. 1.

We envision a suite of tactics that discover relationships between an entity and an event. These tactics use analytics that can create new relationships. In other words, they can add new triples to the knowledge base. Nearness Predicate belongs to this suite. Nearness Predicate uses an analytic that creates new triples using the near predicate. In other words, it creates triples of the form (P near E) where P is an entity and E is an event.

IV. Overview of TT-IQ

In this section, we give the formal definition of TT-IQ. Due to space limitations, we omit some rigor that would be found in a normal presentation of type theory.

Both types and objects are terms. We define the terms T and t as follows.

$$
\begin{aligned}
T, t \coloneqq\ & \bot \mid s \mid r \mid x \mid X \mid f(t_1, \dots, t_n) \mid t.a \mid t.\mathbf{size} \\
& \mid \mathbf{tt} \mid \mathbf{ff} \mid (a_1 = t_1, \dots, a_n = t_n) \mid [t_1, \dots, t_n] \\
& \mid T \wedge T' \mid T \vee T' \mid T \to T' \mid \neg T \mid P(T_1, \dots, T_n) \mid \mathbf{prop} \\
& \mid \mathcal{S} \mid \mathcal{R} \mid a_1 : T_1 \times \dots \times a_n : T_n \mid T^* \mid \{x : T \mid T'\} \\
& \mid \forall x : T.\, T' \mid \exists x : T.\, T' \mid \forall X \le T.\, T' \mid \exists X \le T.\, T' \mid \hat{X}
\end{aligned}
$$

In the definition of terms, s and r range over strings and numbers respectively; a ranges over attribute names; and X and x range over variables. The term $\bot$ represents null. The terms $\mathbf{tt}$ and $\mathbf{ff}$ represent true and false, respectively. The terms of the form $(a_1 = t_1, \dots, a_n = t_n)$ represent records. Each $a_i = t_i$ in a record represents an attribute where a_i is the name of the attribute and t_i is the value of the attribute. The terms of the form $t.a$ represent selecting the value of an attribute whose name is a from a record t. Terms of the form $[t_1, \dots, t_n]$ represent lists. Terms of the form $t.\mathbf{size}$ represent the number of elements in the list t. P and f range over predicate symbols and function symbols, respectively. $\mathcal{S}$ is the type of strings and $\mathcal{R}$ is the type of numbers. We call the terms s, r, x, $f(t_1, \dots, t_n)$, $t.a$, $t.\mathbf{size}$, $\mathbf{tt}$, $\mathbf{ff}$, $(a_1 = t_1, \dots, a_n = t_n)$, and $[t_1, \dots, t_n]$ *objects*. We call all of the other terms, such as $\{x : T \mid T'\}$ and $\exists x : T.\, T'$, types.

Terms of the form $a_1 : T_1 \times \cdots \times a_n : T_n$ represent record types and terms of the form T^* represent list types. The type $\mathbf{prop}$ represent the type that contains types that represent logical formulas, such as $T \wedge T'$ and $P(T_1, \dots, T_n)$. Any type created using terms in $\mathbf{prop}$ will also be in $\mathbf{prop}$. For example $\mathrm{grt}(5,4) \wedge \mathrm{grt}(7,3)$ is a member of $\mathbf{prop}$. Terms of the form $\{x : T \mid T'\}$ represent set types. Intuitively a set type $\{x : T \mid T'\}$ represents a list of objects of type T where each member of the list makes the type representing the logical formula T' true. We call $\hat{X}$ a *reference*. We use $\hat{X}$ to support injecting terms created by an analytic running in a separate subsystem. Notice there are two kinds of quantifiers, those that range over objects, $\forall x : T$ and $\exists x : T$, and those that range over types, $\forall X \le T$ and $\exists X \le T$. In the quantifiers that range over objects, $x : T$ means x is a member of type T. So, $\forall x : T$ means for all terms x that are members of the type T. In the quantifiers that range over types, $X \le T$ means X is a subtype of T. So, $\forall X \le T$ means for all types X that are subtypes of T.

Notice that types may contain objects. For example, if grt is a predicate symbol that represents greater than equal to and abs is a function symbol that represents absolute zero, then $\{x : \mathcal{R} \mid \mathrm{grt}(\mathrm{abs}(x), 0)\}$ is a type that contains the objects 0 and $\mathrm{abs}(x)$.

Given any two terms t and t' and a variable x, $t[t'/x]$ is the term produced by replacing all free occurrences of x in t with t'. For example, $f(g(x,y),x)[3/x]$ is $f(g(3,y),3)$. For two terms t and T, we write $t : T$ to mean that t inhabits the type T or t is a member of the type T. For example, $3 : \mathcal{R}$ and $[3,4,5] : \mathcal{R}^*$. We give some of the rules of TT-IQ for determining which terms inhabit a type in Fig. 4.

$$\frac{t_1 \rightharpoonup t_1' \quad \cdots \quad t_n \rightharpoonup t_n'}{(a_1 = t_1, \ldots a_n = t_n) \rightharpoonup (a_1 = t_1', \ldots, a_n = t_n')} \text{(REC. } \rightharpoonup\text{)}$$

$$\frac{t \rightharpoonup (a_1 = t_1, \ldots, a_n = t_n) \quad a = a_i}{t.a \rightharpoonup a_i} \text{(FIELD SELECTION } \rightharpoonup\text{)}$$

$$\frac{t_1 \rightharpoonup t_1' \quad \cdots \quad t_n \rightharpoonup t_n'}{[t_1, \ldots, t_n] \rightharpoonup [t_1', \ldots, t_n']} \text{(LIST } \rightharpoonup\text{)}$$

$$\frac{t \rightharpoonup [t_1, \ldots, t_n]}{t.\mathbf{size} \rightharpoonup n} \text{(LIST SIZE } \rightharpoonup\text{)}$$

$$\frac{\Im\big(f(t_1, \ldots, t_n)\big) \mapsto t}{f(t_1, \ldots, t_n) \rightharpoonup t} \text{(FUNCTION APPLICATION } \rightharpoonup\text{)}$$

$$\frac{\Im\big(P(T_1, \ldots, T_n)\big) \mapsto t}{P(T_1, \ldots, T_n) \rightharpoonup t} \text{(PREDICATE APPLICATION } \rightharpoonup\text{)}$$

$$\frac{\Im\big(\hat{X}\big) \mapsto t}{\hat{X} \rightharpoonup t} \text{(REFERENCE } \rightharpoonup\text{)}$$

Fig. 2. Evaluation rules for terms that do not evaluate to themselves.

The specification of an analytic can be generalized as follows. The specification will need to contain universal quantifiers to allow the query engine to pass in types and objects to the analytic. If the analytic takes in m types and n objects, then the specification will need the quantifiers $\forall X_1 \leq U_1. \cdots . \forall X_m \leq U_m$ and $\forall x_1 : T_1. \cdots . \forall x_n : T_n$. We abbreviate these as $\forall \vec{X} \leq \vec{U}$ and $\forall \vec{x} : \vec{T}$, respectively. An analytic may output types and objects. If an analytic generates k types and l objects, then the specification will need to contain existential quantifiers $\exists Y_1 \leq U_1' \cdots \exists Y_k \leq U_k'$ and $\exists y_1 : T_1'. \cdots . \exists y_l : T_l'$. We abbreviate these as $\exists \vec{Y} \leq \vec{U'}$ and $\exists \vec{y} : \vec{T'}$, respectively. The specification will also contain the precondition and the postcondition of the analytic. We denote these respectively as ϕ and ψ. The general term that represents the specification of an analytic is specified in (1).

$$\forall \vec{X} \leq \vec{U}. \forall \vec{x} : \vec{T}. \big(\phi \rightarrow \exists \vec{Y} \leq \vec{U'}. \exists \vec{y} : \vec{T'}. \psi\big) \tag{1}$$

In practice, the number of inputs and outputs of an analytic will be small. For example, the specification of Geo Search, an analytic that finds objects that occur within a region, is as follows.

$$\forall X \leq \{\text{location:KML}\}. \forall r : \text{KML}.\mathbf{true} \rightarrow \exists e : X^*. \forall t : e.\text{inRegion}(r, t) \tag{2}$$

In the specification of an analytic, the X's and Y's in (1) represent concepts and the x's and y's represent individuals. An analytic may also take relationships as input and output relationships. The relationships an analytic takes as input are defined by ϕ, and the relationships an analytic produces are defined by ψ. For example, the Geo Search analytic outputs a relationship $\text{inRegion}(r, t)$. The domain is defined by the type of r which is KML and the range is all subtypes of {location:KML}. Geo Search doesn't take any relationships as input.

Traditionally, the definition of a type theory includes rules for evaluating terms. As a result, most type theories in the literature are definitions of statistically typed functional programming languages. TT-IQ has rules for evaluation. We list a subset of the rules in Fig. 2. TT-IQ should not be considered a functional programming language. TT-IQ does not contain a construct for creating functions. Instead, TT-IQ defines a means for injecting functions and relations into it that are executed by an external subsystem or programming language. We represent the external subsystem or programming language as an *interpreter*. For any term t, an interpreter maps t to a term t'. We use $\Im$ to denote an interpreter. We write $\Im(t) \mapsto t'$ to mean that $\Im$ interprets t as t'. The interpreter is used to evaluate the application of function symbols and predicate symbols. For a term t, we write $t \rightharpoonup t'$ to mean t evaluates to t'. The rules FUNCTION APPLICATION $\rightharpoonup$ and PREDICATE APPLICATION $\rightharpoonup$ indicate that the application of a function symbol or a predicate symbol to a sequence of terms is equal to the interpretation of the application of the function symbol or predicate symbol. These rules do not require that the arguments of the function symbol and predicate symbol be evaluated before passing them to the interpreter. As a result, the interpreter can determine the mode of evaluation, such as lazy evaluation or eager evaluation. The evaluation rule REFERENCE $\rightharpoonup$ illustrates the ability of the interpreter to manage storage of instance data. Intuitively REFERENCE $\rightharpoonup$ means that a reference to a concept evaluates to the set of individuals in the concept. The concept is defined by the type T and $\hat{X}$ is a pointer to an index, graph or other data structure that contains members of the concept.

A paramount feature of TT-IQ is subtyping. TT-IQ uses a unique feature to typing that is required for semantic modeling. Traditionally, subtyping on record types is the same as class inheritance. In other words, a record type T is a subtype of T' if all the attributes in T' are also attributes in T. This means that for every attribute named a in T' there is attribute a in T. However, this definition of subtyping ignores the semantic meaning of attribute names. Two attributes can have different names, but mean the same thing. As a result, TT-IQ defines subtyping so that attribute names do not have to be syntactically the same, but semantically the same. Actually, in TT-IQ attribute names do not need to be equivalent, but one attribute name has to be subsumable by the other. The definition of subtyping in TT-IQ uses a partial ordering on attribute names to define subtyping. Due to space limitations we are unable to define the complete definition of subtyping. We do show the subtyping rule for record types below.

$$\frac{n \geq m \quad \text{for } i = 1, \ldots, m \quad a_i \sqsubseteq a_i' \quad T_i \leq T_i'}{a_1 : T_1 \times \cdots \times a_n : T_n \leq a_1' : T_1' \times \cdots \times a_m' : T_m'}$$

In "Interactive Querying" we showed examples of tactics. A tactic is a program or script that returns a *proof tree.* A proof tree represents a proof of a *type judgment*. Intuitively a type judgment asserts that a term belongs to a type. A type judgment has the form $\Gamma \vdash T : T'$ where Γ represents an *environment* and $T : T'$ represents the assertion that T inhabits

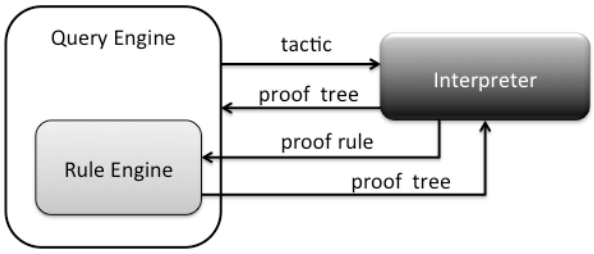


Fig. 3. Interaction between the query engine, the rule engine, and the interpreter.

T'. The environment, Γ, of a type judgment consists of type assignments, terms, predicate symbols and function symbols. It also consists of pairs of predicate symbols and attribute names. These pairs form a partial ordering, $\sqsubseteq$. If $A_1, \ldots, A_n$ are type assignments of terms, predicate symbols, and function symbols, then $\sqsubseteq; A_1, \ldots, A_n$ is an environment.

Only the rule engine can execute a proof rule. The rule engine is a subcomponent of the query engine. Given a rule, a proof tree, and judgment in the proof tree, the rule engine will apply the rule to the judgment. The result will be a proof tree that uses the antecedent of the rule to create children of the judgment. Fig. 3 illustrates the relationship between the query engine, the rule engine and the interpreter.

TT-IQ does not define the language in which tactics are written. An interpreter performs evaluation of tactics. Detailed discussion of the language for creating tactics is outside the scope of this paper. In this paper it suffices to say that a tactic takes as input a typing judgment and outputs a proof tree. The root of the tree has to be the input judgment. The tactic has to use proof rules to create the proof tree.

Fig. 4 contains a few proof rules for TT-IQ. We use $\exists_{\leq}$ ELIMINATION to specify that an analytic will retrieve the individuals of a concept represented as T or a subtype of T. The rule uses a concept reference so that we can postpone selection of the analytic until we have a judgment whose type matches the postcondition of an analytic. We use ANAYLIC EVAL to prove judgments that require an analytic. ANAYLIC EVAL generates concepts and individuals that satisfy a condition specified as a type. The condition can represent a relationship or the search criteria for a query. The condition is specified as the type T in ANAYLIC EVAL. An analytic whose postcondition *matches* T is used to generate the concepts and individuals that satisfy the condition specified as T. In the rule, the analytic is f and its postcondition is ψ. The top-right hypothesis is used to establish that T matches ψ if replacing the free variables in ψ produces a term that is a supertype of T. The terms that replace the free variables are the inputs to the analytic, $\vec{V}$ and $\vec{v}$, and the outputs of the analytic, $\overrightarrow{W}$ and $\vec{w}$. $\vec{V}$ is a sequence of types $V_1, \ldots, V_m$ and $\vec{v}$ is a sequence of objects $v_1, \ldots, v_n$. Likewise $\overrightarrow{W}$ is a sequence of types $W_1, \ldots, W_k$ and $\vec{w}$ is a sequence of objects $w_1, \ldots, w_l$. Intuitively, $\vec{V}$ and $\vec{v}$ are the concepts and individuals required by the analytic to produce the concepts and individuals $\overrightarrow{W}$ and $\vec{w}$ that satisfy the relationship defined by ψ. All free variables in each V_i and v_j are declared in Γ. Each W_i and each w_j do not have any free variables.

Since $\vec{V}$ and $\vec{v}$ represent the input to the analytic, we need to verify they satisfy the precondition ϕ. The three hypotheses of ANAYLIC EVAL on the left achieve this. The top two hypotheses are judgments to verify that the inputs to the analytic have the correct types. The last of the three hypotheses verifies that ϕ is true with all of its free variables replaced with inputs to the analytic. The top hypothesis on the right is used to verify that the postcondition is true for the inputs and outputs of the analytic. The second hypothesis from the top on the right indicates that the output of f on $\vec{V}$ and $\vec{v}$ produces $\overrightarrow{W}$ and $\vec{w}$. Recall from FUNCTION APPLICATION $\rightharpoonup$ in Fig. 2 that an interpreter is used to produce $\overrightarrow{W}$ and $\vec{w}$. The hypothesis on the bottom-right indicates that each W_i evaluates to a list of terms of type Y_i and that each w_j evaluates to a term that is of type T'_j. This hypothesis shows that we intend to represent concepts as list of terms of a specific type. The $\exists_{\{\}}$ REWRITE rule makes use of the fact that for any term t, if $t: \{y: T | T'\}$ then $t: T$ and

$$\frac{\Gamma, \hat{X} \leq T \vdash \phi[\hat{X}/X]:\mathbf{prop}}{\Gamma \vdash \exists X \leq T:\mathbf{prop}} \ (\exists_{\leq} \text{ ELIMINATION})$$

$$\frac{\Gamma \vdash \exists X \leq T.\, (\forall y: X.\, T') \wedge T'':\mathbf{prop}}{\Gamma \vdash \exists X \leq \{y: T | T''\}.\, T'':\mathbf{prop}} \ (\exists_{\{\}} \text{ REWRITE})$$

$$\frac{\begin{array}{cc} \Gamma \vdash \vec{V} \leq \vec{U} & \Gamma \vdash\ T \leq \psi[\overrightarrow{W}/\vec{Y}][\vec{w}/\vec{y}][\vec{v}/\vec{x}][\vec{V}/\vec{X}] \\ \Gamma \vdash \vec{v}:\vec{T} & f(\vec{V}, \vec{v}) \rightharpoonup [\overrightarrow{W}, \vec{w}] \\ \Gamma \vdash \phi[\vec{v}/\vec{x}][\vec{V}/\vec{X}]:\mathbf{prop} & W_i \rightharpoonup t_i \ \text{and} \ \Gamma \vdash t_i: Y_i^* \quad w_j \rightharpoonup t'_j \ \text{and} \ \Gamma \vdash t'_j: T'_j \end{array}}{\Gamma, f: \forall \vec{X} \leq \vec{U}.\, \forall \vec{x}: \vec{T}.\, (\phi \rightarrow \exists \vec{Y} \leq \overrightarrow{U'}.\, \exists \vec{y}: \overrightarrow{T'}.\, \psi), \Gamma' \vdash T:\mathbf{prop}} \ (\text{ANAYLIC EVAL})$$

$$\frac{\Gamma \vdash S \leq T \quad \Gamma \vdash \exists X \leq S.\, T':\mathbf{prop}}{\Gamma \vdash \exists X \leq T.\, T':\mathbf{prop}} \ (\text{NARROW TYPE})$$

Fig. 4. A subset of the type rules of TT-IQ

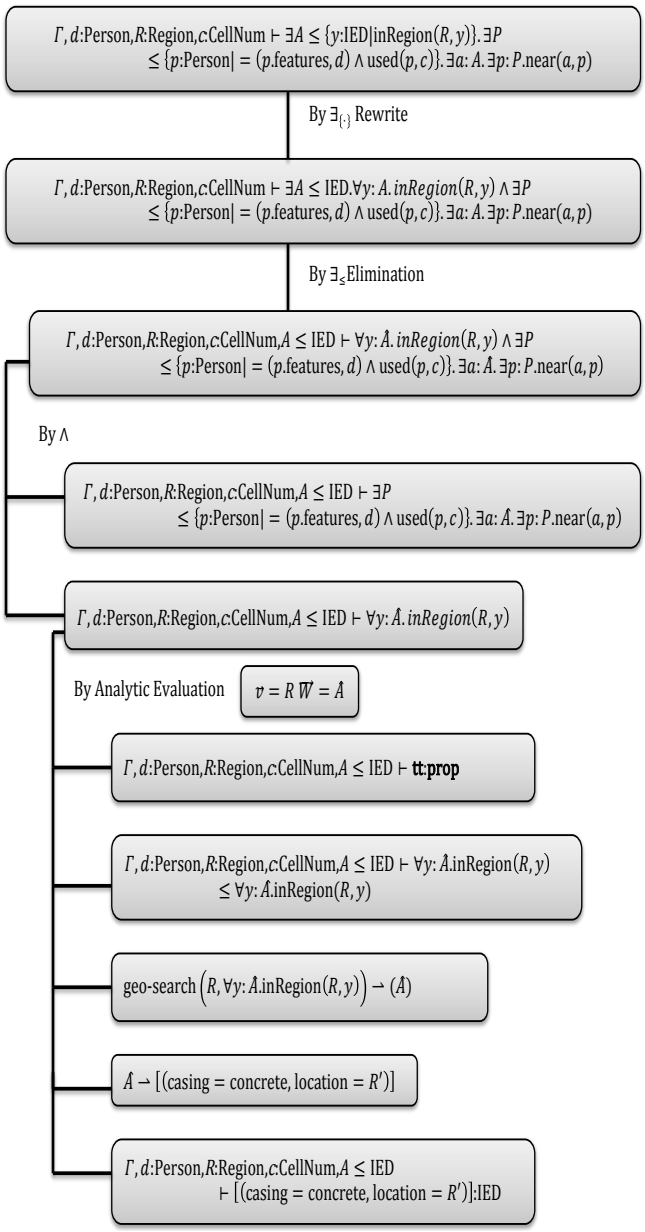


Fig. 5. Proof tree illustrating ∃ rules and analytic evaluation.

$T[t/y]$:**prop**.

Recall in Fig. 1, we used the Narrow Concept tactic to replace Attack with IED. Narrow Concept contains code to select the appropriate concept to use as the subtype for the Narrow Type rule. After it finds a subtype, it asks the rule engine to apply Narrow Type to a target type judgment in a proof tree. The rule engine creates a new judgment using the antecedent of Narrow Type as a template. The new judgment is added as a child of the target type judgment. The rule engine returns the new proof tree to Narrow Concept and Narrow Concept returns the proof tree to the query engine. Fig. 5 contains a portion of the proof tree created from the Geo Search and Concept Introduction tactics in Fig. 1.

V. DISCUSSION

A. *Believability*

A query engine for intelligence data needs to support various levels of believability. We can support believability in TT-IQ by using *type modality [29]*. In other words, we can annotate type with a modality operator that represents a level of believability, such as **certain**, **likely**, **not-likely**, and **impossible**. Then an analyst could prove a hypothesis represented by the type T is likely to occur as the judgment $\Gamma \vdash T^{\mathbf{likely}} : \mathbf{prop}$ or disprove it by proving the judgment $\Gamma \vdash T^{\mathbf{impossible}} : \mathbf{prop}$.

B. *Too complex for an analyst*

The formalism of TT-IQ and the method of reasoning employed by TT-IQ may be too complicated for an analyst. We don't expect the analyst to specify queries in the formal language of TT-IQ, but in natural language similar to that used in Fig. 1. We could employ a technique similar to that used in [14] to allow end users to specify queries using natural language.

C. *Non-determinism and Subtyping*

Since a type may have multiple subtypes, a tactic that finds or creates a subtype of a type could be nondeterministic. In other words, the tactic may not produce the same subtype for the same supertype. As a result, the query engine could produce different results for the same query over the same data. We can resolve non-determinism by asking the user to select the appropriate subtype. This approach would be similar to faceted search. The query engine would require the end user to select from a list of subtypes to use as a candidate to narrow the search space.

D. *Implementation of TT-IQ*

Currently, we are in the planning stage of creating an implementation of TT-IQ. We plan to create an implementation of TT-IQ using Coq and R. Coq will provide the interactive reasoning capability. R will be used as the language and runtime for defining and executing analytics. We anticipate having to add a library to R to provide a more seamless interaction with RDF than the existing R libraries.

We consider this implementation of TT-IQ a proof-of-concept implementation. We plan to use this implementation to conduct research to address usability issues and determine strengthens and weaknesses of interactive semantic querying over automatic semantic querying.

The analyst workstation will contain an analytic framework that would provide an interface to support contribution of analytics written in a wide range of languages, such as MATLAB, C, Java, and Python.

VI. CONCLUSION

In this paper, we showed how to apply techniques from ITPs (interactive theorem provers) to analyze military intelligence. Users of ITPs apply small programs called tactics in an iterative fashion to construct a proof. We demonstrated how tactics could be used to answer semantic queries interactively. Furthermore, we showed how to incorporate

analytics that use machine learning, knowledge discovery, or network analysis into the querying process.

A. Future Work

In "Believability", we eluded to an approach to handle uncertainty. Future work should investigate this approach. Also, we should consider how to incorporate the approach in [15] into our type system.

In the future, we would like to investigate how to implement interactive querying in an existing military intelligence cloud system, such as the DCGS-A Cloud. We believe our approach to querying would be a good fit for the semantic enhancement approach adopted by the DCGS-A Cloud.

ACKNOWLEDGMENT

Rod Moten thanks Andy Zhao of Sotera Defense Solutions and Dr. Aaron Stump of the University of Iowa on their valuable feedback on the ideas expressed in this paper.

REFERENCES

[1] Wylie Wong, "The Army Brings the Cloud to the Battlefield," *FedTech Magazine*, 31-Jul-2013. [Online]. Available: http://www.fedtechmagazine.com/article/2013/07/army-brings-cloud-battlefield. [Accessed: 22-Aug-2013].

[2] S. Miakhel, "Understanding Afghanistan: The importance of tribal culture and structure in security and governance," *Asian Survey*, vol. 35, no. 7, 1995.

[3] B. Ulicny, G. M. Powell, C. J. Matheus, M. Coombs, and M. M. Kokar, "Priority Intelligence Requirement Answering and Commercial Question-Answering: Identifying the Gaps," DTIC Document, 2010.

[4] LTG Keith Alexander, BG Mike Ennis, Robert L. Grossman, James Heath, Russ Richardson, Glenn Tarbox, and Eric Sumner, "Automating Markup of Intelligence Community Data: A Primer," *Defense Intelligence Journal*, no. 12–2, pp. 83–96, 2003.

[5] D. Davis, R. Lichtenwalter, and N. V. Chawla, "Supervised methods for multi-relational link prediction," *Soc. Netw. Anal. Min.*, vol. 3, no. 2, pp. 127–141, Jun. 2013.

[6] D. Davis, R. Lichtenwalter, and N. V. Chawla, "Multi-relational Link Prediction in Heterogeneous Information Networks," in *2011 International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2011, pp. 281–288.

[7] S. David, M. Tatiana, H. Alan, C. Shaun, and S. Barry, "Integration of Intelligence Data through Semantic Enhancement," in *Proceedings of the Sixth International Conference on Semantic Technologies for Intelligence, Defense, and Security*, Fairfax, VA, USA, pp. 6–13.

[8] B. Smith, T. Malyuta, W. S. Mandrick, C. Fu, K. Parent, and M. Patel, "Horizontal Integration of Warfighter Intelligence Data: A Shared Semantic Resource for the Intelligence Community," in *Proceedings of the Seventh International Conference on Semantic Technologies for Intelligence, Defense, and Security*, 2012.

[9] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[10] G. Zenz, X. Zhou, E. Minack, W. Siberski, and W. Nejdl, "Interactive Query Construction for Keyword Search on the Semantic Web," in *Semantic Search over the Web*, R. De Virgilio, F. Guerra, and Y. Velegrakis, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 109–130.

[11] S. Ferré, A. Hermann, and M. Ducassé, "Combining Faceted Search and Query Languages for the Semantic Web," in *Advanced Information Systems Engineering Workshops*, C. Salinesi and O. Pastor, Eds. Springer Berlin Heidelberg, 2011, pp. 554–563.

[12] H. Pérez-Urbina, E. Rodrıguez-Dıaz, M. Grove, G. Konstantinidis, and E. Sirin, "Evaluation of query rewriting approaches for OWL 2," in *Proc. of the Joint Workshop on Scalable and High-Performance Semantic Web Systems (SSWS+ HPCSW 2012)*, 2012, vol. 943.

[13] G. Tecuci, *Building intelligent agents: an apprenticeship multistrategy learning theory, methodology, tool and case studies*. San Diego: Academic Press, 1998.

[14] G. Tecuci, M. Boicu, C. Boicu, D. Marcu, B. Stanescu, and M. Barbulescu, "The Disciple–Rkf Learning and Reasoning Agent," *Computational Intelligence*, vol. 21, no. 4, pp. 462–479, 2005.

[15] M. Boicu, D. Marcu, G. Tecuci, and D. Schum, "Cognitive Assistants for Evidence-Based Reasoning Tasks," in *2011 AAAI Fall Symposium Series*, 2011.

[16] G. Tecuci, D. Schum, M. Boicu, D. Marcu, and Hamilton, Benjamin, "TIACRITIS System and Textbook: Learning Intelligence Analysis through Practice," in *Proceedings of the Fifth International Conference on Semantic Technologies for Intelligence, Defense, and Security*, 2010.

[17] G. Tecuci, D. Marcu, M. Boicu, D. Schum, and K. Russell, "Computational Theory and Cognitive Assistant for Intelligence Analysis," in *Proceedings of the Sixth International Conference on Semantic Technologies for Intelligence, Defense, and Security – STIDS 2011*, Fairfax, VA, USA, 2011, pp. 66–75.

[18] S. F. Allen, M. Bickford, R. L. Constable, R. Eaton, C. Kreitz, L. Lorigo, and E. Moran, "Innovations in computational type theory using Nuprl," *Journal of Applied Logic*, vol. 4, no. 4, pp. 428–469, Dec. 2006.

[19] *Interactive Theorem Proving and Program Development - Coq'Art: The Calculus of Inductive Constructions*. .

[20] T. Nipkow, L. C. Paulson, and M. Wenzel, *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. Springer, 2002.

[21] R. Dapoigny and P. Barlatier, "Using a Dependently-Typed Language for Expressing Ontologies," in *Knowledge Science, Engineering and Management*, H. Xiong and W. B. Lee, Eds. Springer Berlin Heidelberg, 2011, pp. 257–268.

[22] R. Dapoigny and P. Barlatier, "Formal foundations for situation awareness based on dependent type theory," *Information Fusion*, vol. 14, no. 1, pp. 87–107, Jan. 2013.

[23] G. De Giacomo, M. Lenzerini, and R. Rosati, "Higher-Order Description Logics for Domain Metamodeling.," in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011, pp. 183–188.

[24] F. A. Lisi, "A Declarative Modeling Language for Concept Learning in Description Logics," in *Inductive Logic Programming*, F. Riguzzi and F. Železný, Eds. Springer Berlin Heidelberg, 2013, pp. 151–165.

[25] Z. Abedjan and F. Naumann, "Improving RDF Data Through Association Rule Mining," *Datenbank Spektrum*, vol. 13, no. 2, pp. 111–120, Jul. 2013.

[26] C. Benzmüller and A. Pease, "Higher-order aspects and context in SUMO," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 12–13, pp. 104–117, Apr. 2012.

[27] M.-L. Mugnier, "Ontological query answering with existential rules," in *Proceedings of the 5th international conference on Web reasoning and rule systems*, Berlin, Heidelberg, 2011, pp. 2–23.

[28] Moten, Rod, "Using a Type-theoretic Approach to Resolve Heterogeneity in Data Fusion," in *Proceedings of the National Symposium on Sensor and Data Fusion 2012*, Washington, D.C., 2012.

[29] A. Nanevski, F. Pfenning, and B. Pientka, "Contextual modal type theory," *ACM Trans. Comput. Logic*, vol. 9, no. 3, pp. 23:1–23:49, Jun. 2008.

Sketches, Views and Pattern-Based Reasoning

Ralph L. Wojtowicz
Baker Mountain Research Corporation (ralphw@bakermountain.org)
Shepherd University (rwojtowi@shepherd.edu)

***Abstract*—The mathematical theory of sketches provides a graphical framework for describing and relating knowledge representations and their models. Maps between sketches can extract domain-specific context from a sketch, express knowledge dynamics and be used to manage representations created for distinct applications or by different analysts. There are precise connections between classes of sketches and fragments of first-order, infinitary predicate logic. EA sketches are a particular class that is related to entity-attribute-relation diagrams and can be implemented using features available in many relational database systems. In this paper we illustrate sketch theory through development of a simple human terrain model. We apply the theory to an example of aligning sketch-based knowledge representations and compare the approach to one using OWL/RDF. We describe the computational infrastructure that is available for working with sketches and outline research challenges.**

I. Introduction

We use the term *knowledge representation* to refer to a mathematical model of the concepts that we use to understand, reason about and navigate our environment. It evolves in response to new experiences, concept formulation and the mission at hand. Ownership, membership, amicability, people and plans are examples of interrelated entities in this network. We use *decision space* to refer to a sets of individuals and relationships that our knowledge representation organizes. This space is more dynamic, densely populated and uncertain than the knowledge representation. The concept of ownership, for example, encompasses a list of ephemeral connections between individuals and their possessions. Our understanding of ownership persists while instances of this relationship come and go. Moreover, different people can share a common understanding of ownership even if the instances of this relationship that they observe have little or no overlap. They apply the same knowledge representation to distinct models.

Different knowledge representations may characterize the same concept in distinct ways. Renaming the concept 'ownership' as Eigentum or propriété, for example, results in a new presentation of the concept. A complex idea may, more generally, be decomposed into distinct, simpler concepts by different people. Finally, as we build a knowledge model to organize our observations of a greater range of phenomena, we frequently derive and extract parts of it that are suitable for context-based reasoning about particular situations.

A mathematical formulation of knowledge should distinguish between the knowledge representation and decision space models. It should support evolution of the former and the dynamics and uncertainty that are characteristics of the latter. The mathematical framework should support derivation of context-specific views of a knowledge representation and a decision space. Finally, it should provide mechanisms for aligning knowledge models that differ due to simple renaming and more complex reformulations of concepts.

Examples of knowledge representations include relational algebra and its implementation in database languages (such as SQL), entity-relation-attribute diagrams, ontologies, data specifications and sketches. Our interest in the latter results from its graphical nature, deep connections between sketch theory and logic and a rich notion of contextual view of a sketch. Sketch theory has proved to be a valuable tool in mathematical logic and the theory of computer programming languages. Its relationship to other semantic technologies, therefore, warrants further exploration.

The purpose of this paper is to introduce the sketch data model to researchers and practitioners of other semantic-based technologies and to describe a program for its application. We seek to give an overview of the theory through discussion and examples without focusing on the mathematical details. The following themes emerge. (1) An ontology or sketch is a *presentation* of knowledge. Different presentations of the same knowledge are possible. The *theory* of a sketch is the formal mathematical object that such presentations generate. (2) Alignment of distinct knowledge representations and derivation of views of particular ones are more appropriately formulated using theories than presentations. (3) The sketch model emphasizes the distinction between a knowledge representation and its models. Instances, incompleteness and uncertainty may be more appropriately incorporated in models rather than in knowledge representations themselves. (4) The software infrastructure available for working with sketches currently is meager compared to that which has been developed around other semantic technologies such as OWL/RDF.

A. Concept of Operations

Figure 1 illustrates an example concept of operations that shows how the sketch data model might be used in a decision support system. Later in this paper we discuss details of particular aspects of the data pipeline. Data from distributed sources is marshaled into local data models $\mathcal{S}$ that are expressed as sketches. The local sketches are aligned using sketch maps into a common parent $\mathcal{T}$ called a *theory*. $\mathcal{T}$ is the sketch generated by the local sketches taking into account potential overlaps. Parent sketches evolve over time as local ones are modified and new data sources come online. Within a particular mission context, a view $\mathcal{V}$ of the system knowledge representation $\mathcal{T}$ is extracted. The problem of mathematically characterizing the context from event and decision histories is a challenging one and is an active area of research for applications such as Internet search. A *view* is then a sketch equipped with a sketch map into the current parent theory. Models of sketches (including views) are distinct from the sketches themselves.

They include the observed instances that populate the classes and relations that symbols in the sketch represent. Uncertainties and partial information are accounted for in the model, not in the sketch. Data artifacts relevant to a view are analyzed to estimate statistical metrics for potential future states. Figure 1 is conceptual and necessarily incomplete. It does not show, for example, the roles of user interface components, visualization tools and query and reasoning engines, nor of event and decision history archives which would be built into a real command decision system.

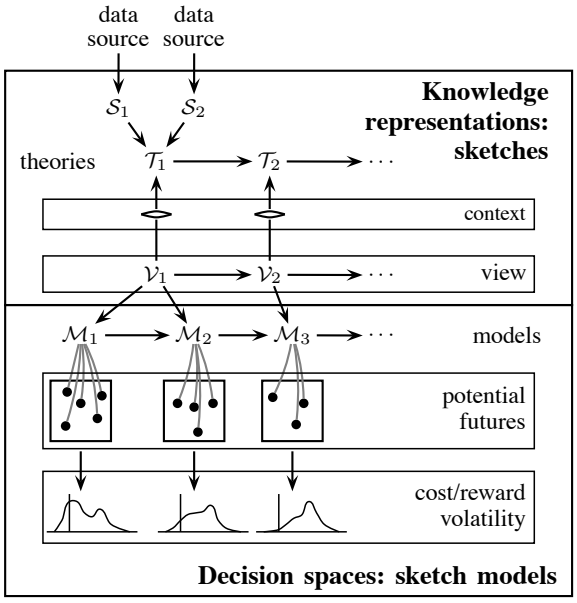


Fig. 1. Concept of operations for the sketch data model

B. Historical Background

Category theory is a mathematical field introduced by Eilenberg and Mac Lane in the 1940s to manage transformations between certain geometric and algebraic structures. It saw explosive growth after Kan discovered the unifying concept of *adjoints* in 1958 [22]. During subsequent decades it has been applied across diverse areas of computer science and mathematics including statistics [9], linguistics [8], dynamic systems, semantics of programming languages [3], topology and, in particular, logic [20] where it provides a non-set-theoretic foundation for mathematics. The theory of sketches is a subdomain of category theory developed by C. Ehresmann in 1968 [11]. It was almost exclusively a tool of the French school of category-theorists until publication of [2], [3]. A category is a collection of *objects* (e.g., sets, probability spaces or vector spaces) and maps between them (e.g., functions, stochastic matrices or linear transformations). In this paper we use categories to construct models of sketches. Sketches themselves form a category having rich structure [13].

II. Sketches, Models and Maps

A sketch is a graph-based knowledge representation. It consists of an underlying directed graph G together with extra structures that impose semantic constraints on models. Figure 2 shows part of the underlying graph of a simple sketch of human terrain knowledge. The vertices Person, Foreign, Coalition, Resident, Village and TribalElement represent classes of entities. Individuals who populate these classes are typically not represented in G (although it is possible to include them explicitly). They instead occur in semantic models of the sketch and may be realized as, for example, rows in database tables. The SeenIn vertex represents a relation between the two classes to which it has edges. It represents the situation in which foreigners may be observed in one or more local villages. Instances of this relation, like individuals who populate the classes, occur in models of the sketch instead of being represented in the sketch itself.

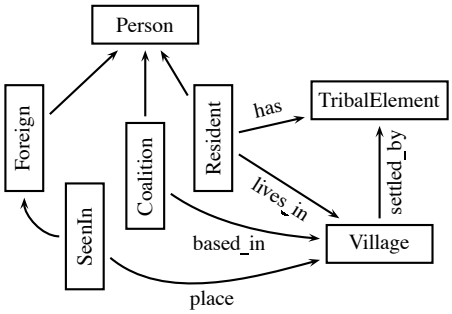


Fig. 2. Part of a graph representing human terrain knowledge

Intuitively, the edges of G represent functions. As we discuss below, however, edges may model incomplete or uncertain information. We intend the edge from Resident to Village, for example, to model a situation in which each resident of an area of interest is associated with a unique home village. Each instance of the class represented by the Coalition vertex is to be based in a specified village. Moreover, each village is settled by a unique tribal element. The two edges from SeenIn represent the process of identifying a foreigner and a village in which he or she was observed. Multiple or no observations of a particular individual are possible. Note that in a sketch, relations (properties) are modeled by vertices rather than edges as is the idiom in OWL/RDF. A relation vertex, however, is the domain of edges that specify the types of entities that it links. That is, the types of the variables that occur as the domain and range of (binary) relations must be specified.

Various features of the human terrain that we seek to represent are not captured by the graph alone. We express these using extra structures called *diagrams*, *cones* and *cocones*. In Section II-A below we give a general discussion of the way in which these constraints are specified using graph maps. In this paper we describe examples of how these concepts are used but do not define them precisely. For details, see [2].

The triangle involving Resident, Village and TribalElement is an example of a *diagram*. It expresses the intuition that the tribal element of a resident R should coincide with the tribal element that has settled the village in which R lives. This semantics is imposed on models of the sketch by including an appropriate diagram in the sketch constraints and by the mathematical definition of sketch model. The constraint can be implemented, for example, using database triggers if the instances are stored in database tables.

The Foreign, Coalition and Resident classes are to be construed as subclasses of Person. Again, this intent is not captured by the graph alone. We express subtype relations by

including particular *cone* constraints in our formulation of the sketch. One such cone would be included for each of the three subtypes that occurs in Figure 2. Cones, like diagrams and cocones, impose mathematical requirements on models.

Finally, we may intend the classes Foreign, Coalition and Resident to be mutually exclusive and to exhaust the possible classifications of Person instances. This feature is not captured by the graph but can be included in the sketch using three cones (to assert the subtype constraints) and a *cocone* to assert the disjoint union constraint.

A. Sketch Maps

A *map* $H \to G$ from a graph H to a graph G is a pair of functions that assigns a G vertex to each H vertex and a G edge to each H edge in a way that respects the source and target information for edges in the two graphs. Graph maps play important roles in defining and applying sketches. First, each of the three types of semantic constraints (diagrams, cones and cocones) is defined as a type of graph map from a base graph B to the underlying graph G of the sketch.

$$B \longrightarrow G$$

The three classes of constraints are distinguished by the shapes of their base graphs. Second, maps between sketches are defined to be maps between the underlying graphs that preserve the semantic constraints. To illustrate this idea, observe that a graph map

$$G \longrightarrow G'$$

between the underlying graphs of two sketches $\mathcal{S}$ and $\mathcal{S}'$ converts each $\mathcal{S}$ constraint $B \longrightarrow G$ into a graph map

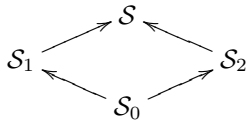
$$B \longrightarrow G \longrightarrow G'$$

via composition of graph maps. If $G \to G'$ is a sketch map, then this composite is also an $\mathcal{S}'$ constraint. Maps between sketches give a rigorous, general framework for addressing knowledge model dynamics, alignment and views. For example, if a knowledge model $\mathcal{S}'$ subsumes another model $\mathcal{S}$, we can express this fact using a sketch map.

$$\mathcal{S} \longrightarrow \mathcal{S}'$$

Not every sketch map expresses a parent-child relationship, however. Those that do are called monomorphisms and satisfy a condition that generalizes the notion of a one-to-one function. A sketch map can, alternatively, merge distinct vertices or edges to eliminate redundancy such as equivalent classes that have been given distinct names.

Alignment of intersecting sketches $\mathcal{S}_1$ and $\mathcal{S}_2$ in a common parent $\mathcal{S}$ is expressed by the following diagram of sketch maps



where $\mathcal{S}_0$ is a sketch representing the intersection of the two knowledge models. We discuss an example in Section II-E.

B. Semantic Models of Sketches

Individuals who populate the classes of a sketch are not typically represented in the sketch itself. They are elements of *models* of the sketch. As we discuss below, this framework clarifies the distinction between the syntax of a knowledge representation and its semantics. We can use this formulation to introduce partiality (i.e., missing data) and uncertainty into models rather than requiring these features to be part of the syntax. First, however, we describe deterministic, set-based models. A (set-based) *model of a graph* G is an assignment of a set $M(v)$ to each vertex v of G and a function $M(e) : M(A) \to M(B)$ to each edge $e : A \to B$ of G. There are no further restrictions on models of a graph.

A *model of a sketch* $\mathcal{S}$ is a model of its underlying graph G that satisfies the restrictions that are represented by the constraints (diagrams, cones and cocones) of $\mathcal{S}$. In this paper we seek to give an overview of the theory and do not give a precise definition of these constraints or their semantics. For details, see [2]. Figure 3 shows a model of a fragment of the human terrain sketch that was shown in Figure 2. The Resident and TribalElement vertices are interpreted as sets of instances. The edge labeled 'has' is interpreted as a function between the sets. To constitute a model of the sketch, the functional interpretations of the edges lives_in and settled_by must be consistent with that of 'has'.

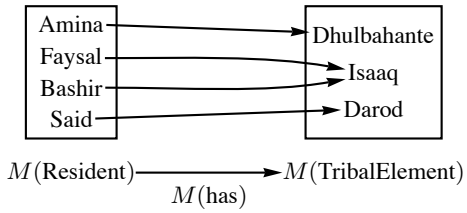


Fig. 3. Functional model of a fragment of the sketch shown in Figure 2.

By varying the semantic category in which sketch models take their values, we may represent lack of information and uncertainty. The edges of the underlying graph may, for example, represent partial functions rather than total functions. Recall that a partial function from a set X to a set Y is a function $X' \to Y$ for some subset of X and that composition $g \circ f$ of partial functions is associative (like composition of total functions) and is defined by further restricting the domain of f. Figure 4 shows a partial function model of a fragment of the human terrain sketch that was shown in Figure 2. In this example, the tribal element membership of Faysal is unknown.

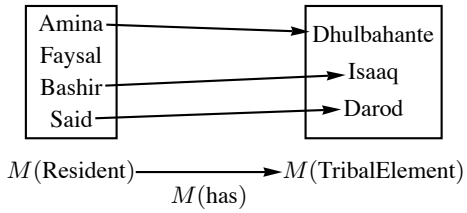


Fig. 4. Partial functional model of a fragment of the Figure 2 sketch.

In Figure 5 we illustrate a probabilistic model of the edge 'has' that occurred in Figure 2. In this model, each point of the source object (which for 'has' is the set M(Resident))

is mapped to a probability function on the target object (the set $M(\text{TribalElement})$ in this case). That is, in this semantic category, edges are interpreted as stochastic matrices (all entries are non-negative and columns sum to 1). Composition is matrix multiplication.

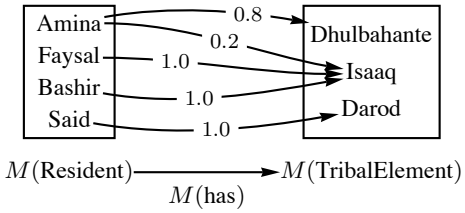


Fig. 5. Probabilistic model of a fragment of the Figure 2 sketch.

There is a rich literature investigating classes of sketches, as distinguished by the types of semantic constraints they include, and classes of semantic models. Examples include linear, finite product, finite limit, EA (entity-attribute) and mixed sketches. The expressiveness of the class of sketch imposes requirements on the classes of structures that may be employed in semantic models. Just as various OWL dialects are associated with different fragments of the predicate calculus, so too are classes of sketches.

C. Maps of Models

The theory of sketches also provides a notion of maps between semantic models. We call these *model maps*. They can be used to represent model dynamics, comparisons and combinations. For example, the fact that different people may populate our tabulations of the Resident class that is represented by the corresponding vertex of Figure 2, should not require us to change the syntax of our knowledge representation. In other words, our understanding of the concepts and relationships of the human terrain does not necessarily change when we observe a new individual to add to our information system. This modular approach to information and knowledge management is a strength of the sketch framework.

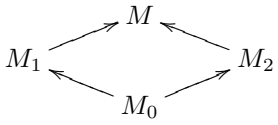
As with models themselves, model maps can introduce partiality and uncertainty. We focus on deterministic maps. Let M and M' be models of a sketch $\mathcal{S}$. For each vertex v of $\mathcal{S}$, the models have corresponding sets $M(v)$ and $M'(v)$ of individuals. A map τ from M to M' is a collection of functions

$$M(v) \xrightarrow{\tau_v} M'(v)$$

between these sets of instances. In order to be a map of models, these functions must be consistent with the functions in the models themselves that arise from edges in the underlying sketch graph. Two models M and M' of the Figure 2 sketch, for example, each have associated sets of Resident and Village instances. If M' subsumes M by, for example, adding new residents, then the new model should maintain the data about the previously-known residents. This is expressed by the following diagram in which we use τ to denote the two functions τ_{Village} and τ_{Resident}.

$$\begin{array}{ccc} M(\text{Resident}) & \xrightarrow{\tau} & M'(\text{Resident}) \\ {\scriptstyle M(\text{lives\_in})}\downarrow & & \downarrow{\scriptstyle M'(\text{lives\_in})} \\ M(\text{Village}) & \xrightarrow[\tau]{} & M'(\text{Village}) \end{array}$$

The definition of map between models requires the two paths to define the same function. That is, the village of a resident who occurs in both models should be the same in both models. Of course, not every map of models represents an extension or subsumption relationship. As with alignment of sketches, we can express alignments of models using maps. For example, alignment of intersecting models M_1 and M_2 in a common parent M is expressed by the following diagram of model maps where M_0 is a model representing the intersection.



D. Presentations and Theories

A sketch (or an OWL ontology) is a compact *presentation* of the much larger body of knowledge $\mathcal{T}$ that it generates. For example, if an ontology defines a class A, a subclass A' of A and a property P that is defined on A, then we can derive a subproperty P' by restricting P to A'. This restriction P' may or may not be explicitly defined in the ontology. It is part of the larger body of knowledge $\mathcal{T}$ that the ontology is designed to present. Sketch theory defines and provides tools for analyzing this generated body of knowledge.

The *theory* $\mathcal{T}$ of a sketch $\mathcal{S}$ is the sketch that $\mathcal{S}$ generates by recursive application of the constructions supported by the type of sketch. These constructions can include property chains (i.e., composition), property inverses (i.e., reciprocals), property restrictions, products (ordered pairs) and coproducts (unions) of classes, and extraction of subclasses and subproperties. The constructions are specified as types of diagrams, cones and cocones since these are the concepts used to specify semantic constraints in sketches. $\mathcal{T}$ is usually much larger than $\mathcal{S}$. It can be infinite even if $\mathcal{S}$ is finite. Consequently, when we write down a knowledge representation, we almost never write down $\mathcal{T}$. We formulate a presentation $\mathcal{S}$ instead.

In Figure 6 we compute a small example. The underlying graph G of the sketch has two vertices and two edges. To make the example a bit more concrete, assume that P represents a class of people and E represents a class of elected officials who serve political districts. The edge r represents an assignment of elected officials to people while u identifies elected officials as particular instances of people. We impose one semantic constraint: the property chain (composite $r \circ u$) of the two edges indicated in the triangle should coincide with the identity function on elected officials. That is, each elected official serves his or her own political district. The finite graph G could, potentially, generate an infinite family of property chains: $r \circ u$, $u \circ r$, $u \circ r \circ u$, $r \circ u \circ r$, etc. The semantic constraint has the effect of truncating this list so that the only distinct properties are those shown in the path graph on the right side of Figure 6. The path graph is the underlying graph of the theory $\mathcal{T}_1$ of the sketch $\mathcal{S}_1$ whose underlying graph is shown on the left side of Figure 6 and whose only constraint is the diagram shown in the center. The derived edge $u \circ r$ connects each person to his or her elected representative.

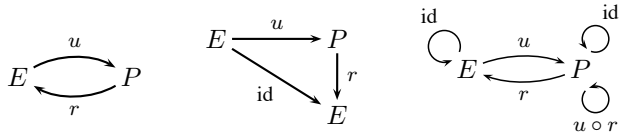


Fig. 6. The two-vertex graph G (left) together with the diagram D (center) generate the theory $\mathcal{T}_1$ (right). The graph and diagram form a sketch $\mathcal{S}_1$.

E. Alignment

A body of knowledge may be presented in different ways even within a fixed formalism (e.g., ontologies or sketches). People use terms differently and use different words to describe the same concepts. The notion of theory of a sketch provides a framework for formulating the alignment problem. Consider the elected officials example discussed above. An alternative presentation is shown in Figure 7. This sketch has only a single vertex C that represents a class of citizens. It has a single edge e that represents the connection of each citizen to his or her elected official. The semantic constraint (indicated by the center diagram) again asserts that each elected official represents himself or herself. The theory generated by this sketch has one vertex and two edges. It is shown below right.

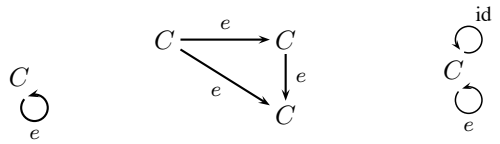


Fig. 7. An alternative formulation of the knowledge model shown in Figure 6. The one-vertex graph G (left) together with the diagram D (center) generates the theory $\mathcal{T}_2$ (right). The graph and diagram form a sketch $\mathcal{S}_2$.

We seek to align these two formulations of the same concepts. To do this we can not use the presentations $\mathcal{S}_1$ and $\mathcal{S}_2$ themselves. We must use theories. Although we can identify the vertex P of the $\mathcal{S}_1$ with the vertex C of $\mathcal{S}_2$, the problem is that there is no edge in $\mathcal{S}_1$ that corresponds to the edge e of $\mathcal{S}_2$. The appropriate edge occurs in the theory $\mathcal{T}_1$ of $\mathcal{S}_1$ not in the sketch $\mathcal{S}_1$ itself. Figure 8 illustrates how the alignment problem is formulated using sketches. The task is to find a sketch $\mathcal{V}$ and sketch maps into the theories generated by the two presentations that we seek to align. $\mathcal{V}$ can be construed as the overlap between the two theories. It is a view (as defined in the next section) of both presentations.

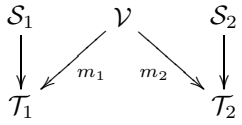


Fig. 8. Formulation of the alignment problem using sketches. To align presentations $\mathcal{S}_1$ and $\mathcal{S}_2$ (i.e., sketches or ontologies), we compute a sketch $\mathcal{V}$ and maps m_1 and m_2 into the theories generated by the presentations.

The sketch framework supports an operation called *pushout* which is essentially the union accounting for overlaps between sketches. With this union operation we can align the two presentations that we have been discussing into a single knowledge representation $\mathcal{T}$. Figure 9 shows the resulting

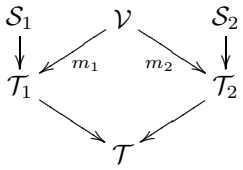


Fig. 9. Alignment of the presentations (e.g., sketches or ontologies) $\mathcal{S}_1$ and $\mathcal{S}_2$ into the common knowledge representation $\mathcal{T}$ using the union (pushout) operation on sketches.

sketches and maps. This illustrates a particular case of the data alignment step shown at the top left of Figure 1.

The need to use structures generated from the available presentations, rather than using the presentations alone, is evident in the alignment example discussed in Chapter 10 of [14]. In this example, two OWL ontologies are aligned using OWL statements. The first ontology is defined below.

```
ex1:Mother       rdfs:subClassOf       ex1:HomeDweller.
ex1:Father       rdfs:subClassOf       ex1:HomeDweller.
ex1:Son          rdfs:subClassOf       ex1:HomeDweller.
ex1:Daughter     rdfs:subClassOf       ex1:HomeDweller.
ex1:hasChild     rdf:type           owl:ObjectProperty.
ex1:hasSon       rdfs:subPropertyOf      ex1:hasChild.
ex1:hasDaughter rdfs:subPropertyOf      ex1:hasChild.
```

The second is defined with prefix `ex2`. It overlaps with but is visibly not equivalent to the first.

```
ex1:Relative     rdf:type                  owl:class.
ex1:Mother       rdfs:subClassOf         ex1:Relative.
ex1:Father       rdfs:subClassOf         ex1:Relative.
ex1:Child        rdfs:subClassOf         ex1:Relative.
ex1:hasParent    rdf:type          owl:ObjectProperty.
```

We align the two using the OWL statements below.

```
ex1:Mother   owl:equivalentClass         ex2:Mother.
ex1:Father   owl:equivalentClass         ex2:Father.
ex1:Son      rdfs:subclassOf              ex2:Child.
ex1:Daugher  rdfs:subclassOf              ex2:Child.
ex1:hasChild owl:inverseOf             ex2:hasParent.
```

Although the `Mother` and `Father` classes coincide, there are no appropriate classes in `ex2` to identify with `ex1:Son` or `ex1:Daughter`. The classes occur in a knowledge representation generated by `ex2` not in `ex2` itself. Similarly, `ex1:hasChild` and `ex2:hasParent` have no corresponding element in the other's ontology. They are identified with elements constructed from the other.

F. Contexts and Views

Decision making uses both general knowledge and specifics of the decision space to balance the expected costs and risks of a program of actions. It focuses on the components that are most relevant to a mission, its goals and tasks. It must efficiently and effectively manage the available data.

Context carries information about intent. Views are implementations of context in a knowledge representation. A *view* of a database is derived using a query. In SQL implementations, a view is typically a single (virtual) table. The *view update problem* addresses the question of how to determine an appropriate update to the state of the total database when a view

is modified. The influential paper [1] developed the *constant complement* approach to view updates. [4], [12] defined the notion of *lens* that characterized a class of updates. At about the same time, [15] described *update strategies* for particular update classes. Sketches were introduced into the study of data semantics in order to better understand database dynamics; in particular, the view update problem [24]. [19] uses sketches to extend the lens concept and classes of view update strategies.

Within the sketch data model, views are particular sketch maps and, in general, are much more expressive than a single derived table. Specifically, a view of a knowledge representation $\mathcal{S}$ is a sketch $\mathcal{V}$ together with a sketch map

$$\mathcal{V} \longrightarrow \mathcal{T}$$

where $\mathcal{T}$ is the theory generated by $\mathcal{S}$ (see Section II-D). Consider, for example, the human terrain sketch shown in Figure 2. One view of interest is obtained by restricting the SeenIn relation to the subrelation in which the village is one in which coalition personnel are based. This view involves subclasses of each of the classes (in addition to the subrelation of SeenIn that we mentioned). None of these subclasses occur in the sketch itself but they are generated by applying cone and cocone constraints.

A challenge to implementing this framework in a decision-making context is using the available event history and other context-specific data to construct an appropriate sketch view in a semi-autonomous manner. The graphical nature of sketches may facilitate the adaptation of recent techniques developed for context sensitive Internet search [6], [16], [21], [28] that use the graph structure of the Web.

G. Support for n-ary Relations

A limitation of OWL/RDF described by practitioners is its lack of direct support for representing n-ary relations. Such properties can be represented directly in sketches. To express a relation R that may hold among n entities that have types $A_1, \cdots, A_n$, we introduce vertices for each of these $n+1$ classes and we include n edges $R \to A_i$. Any axioms that the relation is intended to satisfy would then be formulated using diagrams, cones and cocones.

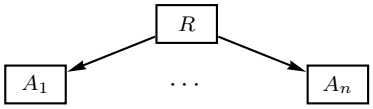


Fig. 10. Sketch for an n-ary relation R among entities of types $A_1, \ldots, A_n$

III. Logical Inference

The graphical nature of the sketch data model supports implementation of pattern-matching reasoning capabilities that emulate the process of experienced decision makers [23]. In classical logic, we express properties and relationships as terms and formulas that are recursively-constructed from basic components. Inference is formulated as rules for deriving valid expressions. Like models of physical phenomena, logics are developed with varying levels of fidelity based on their intended applications. Examples include classical, descriptive, modal and linear logics. Expressiveness, however, comes at the expense of higher computational complexity: inference for first-order logic is undecidable, NP-complete for propositional logic, and P-complete and linear for propositional Horn logic (a property exploited in the Prolog language) [26].

A sketch is an alternative, graphical way of presenting a logical theory [2], [20]. In the sketch data model, we express relationships using diagrams, cones and cocones in directed graphs instead of with formulas and terms. Logical inference employs graph properties associated with constraints. A sketch with no constraints is like a logical signature with no axioms.

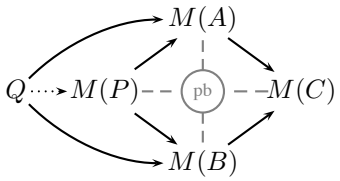


Fig. 11. Universal mapping property that characterizes pullback cones

A pullback cone, for example, is characterized by the property illustrated in Figure 11 which shows a model M of such a constraint (see [2]). If the two outer functions from Q to $M(C)$ are equal, then there is a unique function from Q to $M(P)$ for which the two paths from Q to $M(A)$ are equal as are the two from Q to $M(B)$. Such graph definitions are called *universal mapping properties* [22]. From these we derive other inference rules such as: If the function from $M(B)$ to $M(C)$ is a subtype (`is_a`) relationship, then so is the edge from $M(P)$ to $M(A)$ [22].

Sketches, like logics, are developed with varying levels of fidelity. Linear sketches are the least expressive. Finite limit, finite sum, EA (entity-attribute) and mixed sketches are richer. Despite the distinct character of logical and sketch-based inference, they share deep connections. For various classes of sketches, there are algorithms for constructing logical theories that have equivalent categories of models (see D.2.2 of [20]). Reasoning about a knowledge model expressed as a sketch, therefore, may be achieved either directly using the computational category theory techniques discussed below in IV or indirectly by converting to a first-order theory and using a predicate calculus reasoner.

The problem of pattern-based reasoning with sketches is similar to the ontology alignment problem [18] that is solved mathematically via the *theory* (or *syntactic category*) of a sketch [20]. We may align, for example, the human terrain sketches that are shown in Figures 13 and 2 via a sketch map from the latter to the former. Refinement of the knowledge base to represent levels in a tribal hierarchy (e.g., ethnic groups, tribes, clans and factions) is accomplished with a sketch map from the Figure 2 sketch to a new sketch that would include additional edges and constraints. The simple business knowledge representation shown in Figure 12 can be mapped to a sub-sketch of our human terrain model.

IV. Software Infrastructure

The software infrastructure available for working with sketches is meager relative to that associated for other semantic models (e.g., OWL/RDF). The Easik tool[1] is the most mature.

[1] http://mathcs.mta.ca/research/rosebrugh/Easik

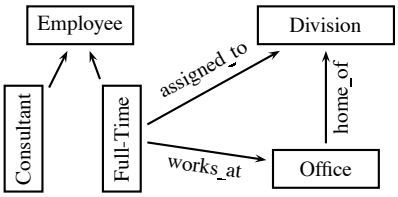


Fig. 12. Part of a sketch representing business structure knowledge

It provides a graphical interface for building a collection of sketches and views. It implements procedures for reading and writing sketches to and from XML and SQL files. It also provides an interface to models maintained in PostgreSQL and MySQL databases. Easik does not implement a reasoning engine. Figure 13 shows a sketch that is similar to the one whose graph is shown in Figure 2. It was developed using Easik. The screen shot illustrates convenient abbreviations for various semantic constraints. The decorated arrows to Person, for example, indicate subtype relationships. These are implemented as cones. Vertices connected to the + symbol form a kind of cocone. Its base consists of the vertices Coalition, Foreign and Resident. The paths connected to CD indicate a diagram constraint.

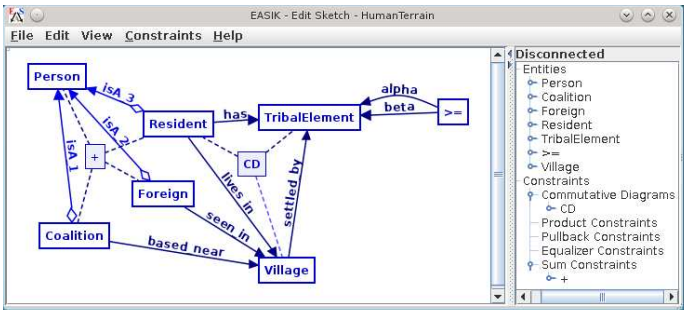


Fig. 13. Human terrain sketch implemented using the Easik software tool

Category theory, despite its abstract nature, is highly computational. All semantic constraints, for example, can be computed from four basic types: cones can be expressed using product cones and equalizer cones; cocones can be expressed using coproducts and coequalizers. One freely-available implementation of these and related computations is written in the ML programming language and is described in [25]. This work and similar research activities could provide a basis for implementing a reasoning engine for a sketch-based information system.

If the only constraints of the sketch are diagrams (i.e., the sketch has neither cones nor cocones), then the theory $\mathcal{T}$ (when $\mathcal{T}$ is, in fact, finite) generated by the sketch may be computed via the left Kan extension algorithm which generalizes the Todd-Coxeter procedure from group theory [7], [27]. If the generated sketch is infinite, the algorithm, of course, does not terminate. Its complexity in the cases when it does terminate has not been characterized and is highly sensitive to small variations in the sketch [5].

V. Implementing Sketch Models in Databases

Features available in major relational database systems including PostgreSQL and MySQL provide an interface between the mathematical theory of sketches and their application. The Easik tool supports read and write operations between knowledge representations and XML files, SQL files and SQL (MySQL or PostgreSQL) database connections. In Easik, a sketch is implemented as a *database schema*. Each sketch entity (graph node) is a *table* created according to the schema. Values that populate the tables form a model of the sketch. Each table has an implicit, integer-valued primary key. In general, a primary key constraint on a table expresses the fact that the values in one or more columns together are a unique identifier of a row. This does not preclude the possibility that two rows may refer, for example, to a single individual. Attributes are table columns. For example, an entity B with no attributes or outgoing edges is implemented in PostgreSQL as follows where the `id` column is an automatically-generated (i.e., `SERIAL`), key.

```
CREATE TABLE B ( id SERIAL PRIMARY KEY );
```

A *foreign key* constraint specifies that the values in one or more columns must match the values occurring in some row of another table. We implement a sketch edge $A \xrightarrow{e} B$ as a foreign key contained in the A-table and referencing the primary key of the B-table. In PostgreSQL this is expressed as follows.

```
CREATE TABLE A ( id SERIAL PRIMARY KEY,
                 e INTEGER NOT NULL REFERENCES
                 B (id) ON DELETE CASCADE
                        ON UPDATE CASCADE );
```

Insertion of a row into the A-table, therefore, involves specifying values for the columns that are introduced into that table for the edges having domain A. Deletion of a B-row can impact the A-table if an A-row references the B-row via the edge $A \xrightarrow{e} B$. Foreign keys serve to implement relations (object properties) in the sketch data model since a relation is simply an entity having edges to the nodes that correspond to the types of its participants.

Although subtype relations (monic edges) are a particular kind of cone constraint, they can be implemented as foreign keys with unique references in the codomain table. In general, however, sketch constraints are implemented using triggers. A *trigger* for a database table or view executes a specified function whenever certain events occur. The simple diagram shown in Figure 14, for example, asserts that semantics of the composite edge f followed by g should equal that of h. In PostgreSQL we express this as follows.

```
CREATE FUNCTION commutativeDiagram0()
      RETURNS trigger AS $commutativeDiagram0$
   DECLARE _cdTarget1 CONSTANT INTEGER := NEW.h;
           _cdTarget2 CONSTANT INTEGER :=
                      (SELECT B.g FROM B
                       WHERE B.id = NEW.f);
   BEGIN   IF   _cdTarget1 IS DISTINCT
                           FROM _cdTarget2
           THEN RAISE EXCEPTION
             'Commutative diagram constraint
              failure';
           END IF;
           RETURN NEW;
   END;
$commutativeDiagram0$ LANGUAGE plpgsql;
CREATE TRIGGER commutativeDiagram0
  BEFORE INSERT ON A
    FOR EACH ROW EXECUTE
      PROCEDURE commutativeDiagram0();
```

It is a trigger that fires before insertion of a row into the A-table to confirm that the values entered in the foreign key columns for f and h satisfy the commutativity constraint taking into account the value in the g-column in an appropriate row of the B-table. Cone and cocone constraints are all similarly implemented. All EA sketch constraints can be constructed from these [22].

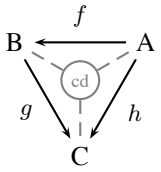


Fig. 14. Diagram to implement using a SQL trigger

A research question that arises is how might one utilize the sketch data model in a context of large-scale, distributed data. Broader demand for a scalable system that supports views and integrity constraints are well-known. Megastore, Tenzing, and Spanner are Google products developed to meet this demand. Apache Cassandra is an open-source alternative.

VI. Conclusion

OWL/RDF and related semantic web technologies have established tenable positions in the intelligence, defense and security domains. The sketch data model, however, integrates a variety of features that can be leveraged. These include its deep connections with infinitary predicate logic, the ability to implement sketches and their models using major relational database systems, its graphical nature and, perhaps most significantly, its sophisticated notion of view of an information system. It is possible to formulate OWL constructs using sketches. The classes of sketches that can be expressed in the dialects of OWL2 is an open question. We have illustrated these concepts and their application to a simple ontology alignment problem.

The sketch data model also clarifies the formulation of certain challenges that we encounter in applications of OWL/RDF. In the sketch approach, uncertainty and lack of information are aspects of models of the sketch. They are not features of the knowledge representation itself. Moreover, sketches (and OWL ontologies) are more appropriately construed as presentations of the larger bodies of knowledge that they generate. In sketch theory, this larger knowledge base is called the theory of a sketch. Except in simple cases of renaming, alignment of presentations involves maps into theories. The extent to which procedures for generating a theory from a sketch can support partial-automation of alignment problems is an open research question.

Finally, the concept of view of a knowledge representation is formulated as a sketch map to a theory. This generalizes the notion of view of a database. Recent techniques developed for context sensitive Internet search exploit the graph structure of the Web and search histories. The extent to which these techniques and the graphical nature of sketches can be exploited to support semi-automated extraction of context-relevant views of a knowledge representation is another open research challenge.

References

[1] F. Bancilhon and N. Spyratos. Update Semantics of Relational Views. *ACM Transactions on Database Systems*. **6**:557–575. 1981

[2] M. Barr and C. Wells. *Toposes, Triples and Theories*. Springer. 1985

[3] M. Barr and C. Wells. *Category Theory for Computing Sciences*. Prentice-Hall. 1990

[4] A. Bohannon, J. Vaughan and B. Pierce. Relational Lenses: A Language for Updatable Views. In *Proc. of the 25th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. ACM. 2006

[5] J. J. Cannon, L. A. Dimino, G. Havas and J. M. Watson. Implementation and Analysis of the Todd-Coxeter Algorithm. *Mathematics of Computation*. **27**(123):463–490. July 1973

[6] H. Cao, D. Jiang, J. Pei, E. Chen and H. Li. Towards Context-Aware Search by Learning a Very Large Variable Length Hidden Markov Model from Search Logs. In *Proceedings of the 18th World Wide Web Conference (WWW'09)*. pp. 191–200. 2009

[7] S. Carmody, M. Leeming, R. F. C. Walters. The Todd-Coxeter Procedure and Left Kan Extensions. *J. Symbolic Computation*. **19**:459–488. 1995

[8] C. Casadio, P. J. Scott and R. A. G. Seely, Eds. *Language & Grammar: Studies in Mathematical Linguistics and Natural Language*. CLSI Publications. 2005

[9] N. N. Čencov. *Statistical Decisions Rules and Optimal Inference*. American Mathematical Society. 1982

[10] E. F. Codd. *Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks*. IBM Research Report RJ599. 1969

[11] C. Ehresmann. Esquisses et types de structures algébriques. *Bul. Inst. Politehn. Iaşi*. **14**:1–14. 1968

[12] J. Foster et al. Combinators for Bi-Directional Tree Transformations: A Linguistic Approach to the View Update Problem. *ACM Transactions on Programming Languages and Systems*. **29**. 2007

[13] J. W. Gray. The category of sketches as a model for algebraic semantics. In *Categories in Computer Science and Logic*. V. 92 of Contemporary Mathematics. AMS. pp. 109–135. 1989

[14] J. Hebeler, M. Fischer, R. Blace and A. Perez-Lopez. *Semantic Web Programming*. Wiley Publishing, Inc. 2009

[15] S. J. Hegner. An Order-Based Theory of Updates for Closed Database Views. *Annals of Math. and Artificial Intelligence*. **40**:63–125. 2004

[16] T. Joachims. Optimizing Search Engines Using Clickthrough Data. *SIGKDD'02, Alberta Canada*. 2002.

[17] M. Johnson and R. Rosebrugh. Sketch Data Models, Relational Schema and Data Specifications. *Electr. Notes Theor. Comp. Sci.* **61**(6):1–13. 2002

[18] M. Johnson and R. Rosebrugh. Ontology Engineering, Universal Algebra, and Category Theory. In *Theory and Applications of Ontology: Computer Aplications*. Springer-Verlag. pp. 565–576. 2010

[19] M. Johnson, R. Rosebrugh and R. J. Wood. Lenses, Fibrations and Universal Translations. *Mathematical Structures in Computer Science*. **22**:25–42. 2012

[20] P. Johnstone. *Sketches of an Elephant*. Oxford University Press. 2005

[21] A. Kustarev, Y. Ustinovesky and P. Serdyukov. Measuring Usefulness of Context for Context-Aware Ranking. *ACM WWW 2012 Companion, Lyon, FR*. 2012

[22] S. Mac Lane. *Categories for the Working Mathematician*. 2nd Ed. Springer-Verlag. 1999

[23] J. Morrison, R. T. Kelly, R. A. Moore and S. G. Hutchins. Implications of Decision Making Research for Decision Support and Displays. In *Decision Making Under Stress: Implications for Training and Simulation*. J. A. Cannon-Bowers and E. Salas, Eds. pp. 375–406. 1998

[24] R. Rosebrugh and R. J. Wood. Relational Databases and Indexed Categories. *Category Theory 1991: Proceedings of an International Summer Category Theory Meeting Held June 23–30*. Vol. 13 of CMS Conference Proceedings. AMS. pp. 391–407. 1991

[25] D. E. Rydeheard and R. M. Burstall. *Computational Category Theory*. Prentice-Hall. 1988

[26] A. Troelstra and H. Schwichtenberg. *Basic Proof Theory*. Cambridge University Press. 2000

[27] R. L. Wojtowicz and N. S. Yanofsky. *Quantum Kan Extensions and Applications*. DOI Contract D11PC20232 Final Report. 2013

[28] B. Xiang, D. Jiang, J. Pei, X. Sun, E. Chen and H. Li. Context-Aware Ranking in Web Search. *ACM SIGIR'10 29–23 July, Geneva, Switzerland*. 2010

An Ontological Inference Driven Interactive Voice Response System

Mohammad Ababneh
Department of Computer Science
George Mason University
Fairfax, VA, USA
mababneh@gmu.edu

Duminda Wijesekera
Department of Computer Science
George Mason University
Fairfax, VA, USA
dwijesek@gmu.edu

Abstract— **Someone seeking entry to an access controlled facility or through a border control point may face an in person interview. Questions that may be asked in such an interview may depend on the context and vary in detail. One of the issues that interviewers face is to ask relevant questions that would enable them to either accept or reject entrance. Repeating questions asked at entry point interviews may render them useless because most interviewees may come prepared to answer common questions. As a solution, we present an interactive voice response system that can generate a random set of questions that are contextually relevant, of the appropriate level of difficulty and not repeated in successive question answer sessions. Furthermore our system will have the ability to limit the number of questions based on the available time, degree of difficulty of generated questions or the desired subject concentration. Our solution uses Item Response Theory to select questions from a large item bank generated by inferences over multiple distributed ontologies.**

Keywords—Ontology; Semantic Web; OWL; Dialogue; Question Answering; Voice Recognition; IVR; VXML; Access Control Policy; Security; Item Response Theory.

I. INTRODUCTION

Physical control points such as human guarded gates, border control points and visa counters provide entry into facilities or geographical regions to those that can be *admitted legitimately*. Legitimacy is usually determined by rules, regulations or policies known to entry control personnel whose duty is to ensure that these policies are enforced while admitting people. In order to do so, they hold an interview, in which an aspiring entrant is asked a series of questions, and possibly show some documents and demonstrate some knowledge about the contents of the documents or attributes contained in them. Successful interviews should have questions that are relevant, of a reasonable level of difficulty (i.e. not too difficult or common knowledge) and not to have been asked in prior interviews for the same purpose without drawing accusations of bias from rejected entrants. Ideally, a successful interview should accommodate differences in accents and provide assurance that it is unbiased against similar attributes.

Given the recent success of interactive voice response (IVR) systems such as auto attendants, satellite navigation, and personal assistants such as Apple's Siri, Google's Voice, Microsoft's Speech, we investigated the possibility of specializing IVR systems for access control such as: Visa interviews, entry point interviews, biometric enrollment interviews, password reset, etc.

Although IVR systems have come a long way in recognizing human voice, and responding to human requests as if responses come from another human, most of the existing IVR systems are pre-programmed with questions and their acceptable answers, and consequently have limited capability in satisfying the Use Case at hand.

The first minor limitation of current IVR systems comes from the fact that, the human starts and drives the conversation. The second limitation is that most IVR systems have a finite number of pre-programmed conversations. Therefore the set of questions generated by such a system are the same for every conversation. This limitation may expose the set of questions so that aspiring entrants may come with prepared question-answer pairs, even if the subject matter of the questions may be unfamiliar to them. Consequently, having the ability to select questions from a large pool may resolve this limitation. The third limitation is that when selecting a random set of questions from a large pool, the set of questions asked may not have the desired overall level of difficulty to challenge the user. Solving this issue is relevant because all aspiring entrants expect to have a fair interview. The forth limitation is that questions must be able to discriminate between someone that knows the subject matter from someone who guesses an answer.

As a solution we created an ontological inference based IVR system that uses item response theory (IRT) to select the questions [13, 3]. Our system uses the XACML language as a base to establish entry policies that consist of rules to specify the attributes that must be possessed by permitted entrants [7]. The IVR system has the responsibility of determining access by asking questions generated using ontological inferences and IRT.

In previous work, we introduced a policy-based IVR system for use in access control to resources [1]. Later, we presented an enhancement that uses IRT to select queries from a large set of attributes present in a policy [2]. Here we introduce ontology-aided access control system by including questions related to the base attributes in order to ascertain the interviewee's familiarity, and provide a score for the entire set of answers [8]. We also have the added capability to generate the succeeding question based on the accuracy of the preceding question. We do so by aligning each attribute with an ontology that encodes the subject matter expertise on that attribute and derive facts from these ontologies using reasoners to generate questions. We then assign weights to these derivations based on the axioms and rules of derivations used in the proof tree.

Usually ontologies have a large number of axioms and assert even more facts when using reasoners. Consequently, blindly converting such an axiom base to human-machine dialogue would result in very long conversations with many disadvantages. The first is that human users would become frustrated of being subjected to long machine driven interrogations, and thereby reducing the usability of the system. The second is that long conversations take longer time to arrive at an accept/reject decision, and likely to create long queues at points of service, such as Airports and guarded doors. In addition, having a line of people behind one person in close proximity may leak private information of the interviewee. Also, others may quickly learn the set of questions and answers that would get them mistakenly authorized, thereby gaining unauthorized access.

We use IRT, which provides the basis for selecting tests from large number of potential questions. Psychmotricans in social sciences and standardized test preparation organizations such as the Educational Testing Services that administer standardized test examinations like SAT, MCAT, GMAT etc. have developed methodologies to measure an examinee's trust or credibility from answers provided to a series of questions. In traditional tests, the ability of the examinee is calculated by adding up the scores of correct answers. Currently, Computerized Adaptive Testing (CAT) that relies on IRT has been used to better estimate an examinee's ability. It has also been shown that the use of CAT/IRT reduces the number of questions necessary to reach a credible estimation of the examinee's ability by 50%. CAT/IRT can be used to control the number and order of questions to be generated based on examinee's previous answers [4, 5].

Our goal in this work is to demonstrate and build an access control system using dialogues of questions and answers generated from a suitable collection of ontologies. Table I shows a sample dialogue that is generated from our research. Our prototype automated IVR system can help immigration enforcement at a border control point making a decision to permit or deny a person asking for entry. Through a dialogue of questions and answers, the interviewee will be assigned a numerical score that will then serve as a threshold in the decision making process. This score is calculated using IRT, which takes into account the correctness of the user's responses and the weight of the individual questions.
The rest of the paper is written as follows. Section II describes an ontological use case, Section III describes the response theory. Section IV describes the system architecture. Section V describes our implementation. Section VI is about experimental results and section VII concludes the paper.

II. Motivating Use Case

In this section, we describe an example ontology used in our work to generate efficient dialogues of questions

TABLE I. A SAMPLE DIALOGUE

	Question/answer	Source	Answer Correct 1 Wrong 0	IRT difficulty	θ (all correct)	θ (all wrong)
System	What is your name	policy	1	1.0	1.0	1.0
User	Dias Kadyrbayev		*Apply policy*			
System	How old are you?	policy	0	1.0	1.0	1.0
User	19		*Apply policy*			
System	I see you have visited Russia, which part?	Ontology - Geospatial	0	1.0	1.3	0.8
User	Dagestan		*Dagestan is a republic within Russia with troubles*			
System	Was there any terrorist attacks while you were there?	Ontology – World Event	1	1.5	1.9	0.5
User	No		*Check world event ontology (ex BBC news Ontology)*			
System	Do you know Dzhokhar Tsarnaev ?	Ontology - FOAF	1	2.0	2.3	0.1
User	Yes	Foaf:knows	*Friend of Dzhokhar Tsarnaev - Brother of Tamerlan Tsarnaev; the Boston bomber.*			
System	Where is he?	Ontology - Geospatial	1	3.0	2.6	-0.5
User	He is in Georgia		*But which Georgia? Is it the country or the US state?*			
System	Are you still attending UMass Dartmouth ?	Dept. of state Visa database	1	3.0	2.6*	-0.5*
User	Yes		*Check if Visa is still valid*			

and answers that are used in assigning a numerical value to an interviewee's ability or trust level.

Fig. 1 illustrates a class diagram of our under-development ontology for homeland security. The purpose of this ontology is to collect, organize and infer information that can help deterring possible attacks, enforcing strict entry and enabling faster reach to suspects. The ontology defines classes, individuals, properties and relationships using OWL 2 Web Ontology Language (OWL) [9]. The major entities in the ontology are:

- **Person:** defines humans in general and has subclasses like; International Student and Friend.
- **Event:** defines an event that has a location, date, time and type like terrorist attack
- **International Student:** is a person who is on an F-1 or J-1 Visa type
- **University:** defines a university. Some of its current members are MIT and GMU
- **City:** defines a city like Boston
- **Country:** defines a country like USA, Russia, Dagestan, Kazakhstan, etc.
- **State:** defines a state like Massachusetts
- **Visa:** defines visa types like F-1 and J-1 student visas and maybe others.

This ontology represents many kinds of data classes and relationships between these major classes and individuals. For example, we define the "Boston Marathon Bombing" as a "Terrorist Attack" that happened in "Boston", which is a city in "Massachusetts" state. Another fact is that "Dzhokhar Tsarnaev" is an "Event Character" in the "Boston Marathon Bombing" "Terrorist Attack". Also we have an "International Student" who is a friend to "Event Character" in the "Boston Marathon Bombing".

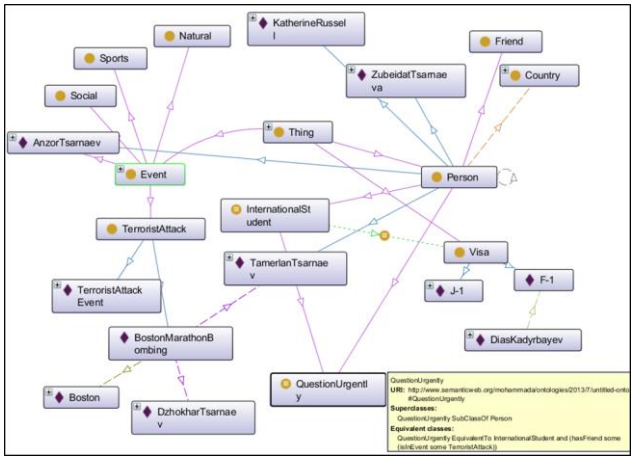


Fig. 1. The Homeland Security Ontology in Protégé

We use this ontology in our work because it serves as a good example showing the strength of our system. First, it shows the possibility of generating valuable questions from asserted or inferred facts. Second, it enables the implementation of the theory under consideration (to be discussed later in the background section) to generate efficient and secure dialogs that are used in: (1) making entry control decisions, (2) assigning numerical values to ability or trust in the shortest time possible and (3) load distribution among interviewers and diverting people for further investigation.

The use of ontology in such an application provides many benefits. The most important amongst them is reasoning. Using a reasoner we are able to derive facts from asserted ones. These facts are used to generate questions to measure the knowledge or ability level of an interviewee on a subject under questioning. In IRT, better item selection and ability estimation happens when a large set of items is available to draw questions from. Using ontology, the large number of derivable facts provides us with the ability to increase the number of questions, and also control the quality and difficulty of questions.

Although there are many reasoners such as FaCT++, JFact, Pellet, RacerPro, we use HermiT [12] in our work. Given an OWL file, HermiT can determine whether or not the ontology or an axiom is consistent, identify subsumption relationships between classes and deduce other facts. Most reasoners are also able to provide explanations of how an inference was reached using the predefined axioms or asserted facts.

One such fact derived from asserted ones in our ontology, is finding the friends that hold a student visa of a person involved in a terrorist attack. To explain this, we have "dzhokhar is friend of Dias", "Dias is friend of Azamat", "Dias has F-1 visa", "Azamat has a J-1 visa", "dzhokhar is an "Event Character" in the "Boston Marathon Bombing", "Boston Marathon Bombing" is a "Terrorist Attack". Thus we infer (using the HermiT reasoner) that Azamat and Dias are the friends of the Boston Bomber and therefore need to be questioned at any entry point. We use this chain of derivations to generate specific questions from them.

Reasoners and the explanations that they provide are very important components in our work to generate relevant and critical questions from ontology that measure knowledge and estimate ability from a response in order to grant access or assign trust. In the example above, the reasoner provided an explanation of the inference using 11 axioms. We use such a number in defining the difficulty of questions generated from such inferences, as

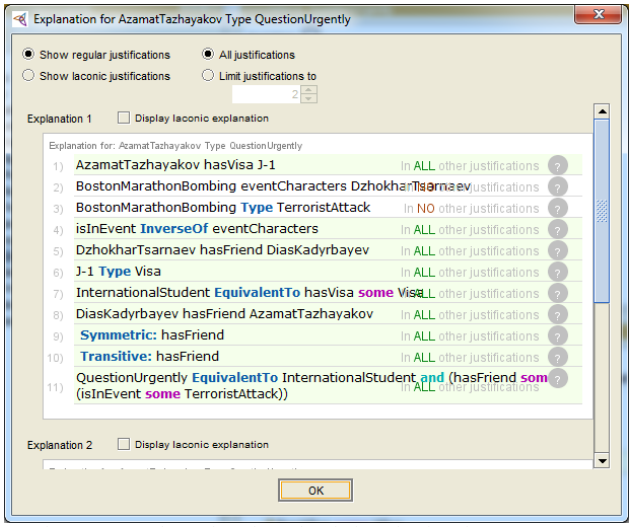


Fig. 2. A sample explanation of an inferred axiom in Protégé using the HermiT reasoner

will be explained in section V. Fig. 2 shows the HermiT reasoner explanation of our inferred fact.

III. BACKGROUND

A. IVR Systems

The main purpose of an IVR system is to interact with humans using a voice stream. An IVR environment consists of a markup language to specify voice dialogues, a voice recognition engine, a voice browser and auxiliary services that allow a computer to interact with humans using voice and Dual Tone Multi-Frequency (DTMF) tones with a keypad enabling hands-free interactions between a user and a host machine [13]. Recently, many applications such as auto attendant, satellite navigation, and personal assistants such as Apple's Siri, Google's Voice, Microsoft's Voice, etc., have started using IVR systems. The IVR language we use is VoiceXML, sometimes abbreviated as VXML [14]. Briefly, Voice XML is a Voice Markup Language (comparable to HTML in the visual markup languages) developed and standardized by the W3C's Voice Browser Working Group to create audio dialogues that feature synthesized speech, digitized audio, recognition of spoken and (DTMF) key inputs, recording of spoken input , telephony, and mixed initiative conversations.

B. Item Response Theory

IRT, sometimes called *latent trait theory* is popular among psychometricians for testing individuals, and a score assigned to an individual in IRT is said to measure his *latent trait* or ability. Mathematically, IRT provides a characterization of what happens when an individual meets an item, such as an exam or an interview. In IRT, each person is characterized by a proficiency parameter that represents his ability, mostly denoted by (θ) in literature. Each item is characterized by a collection of parameters mainly, its difficulty (b), discrimination (a) and guessing factor (c). When an examinee answers a question, IRT uses the examinee's proficiency level and the item's parameters to predict the probability of the person answering the item correctly. The probability of answering a question correctly according to IRT in a three-parameter model is shown in (1), where e is the constant 2.718, b is the difficulty parameter, a is the discrimination parameter, c is the guessing value and θ is the ability level [3].

$$P = c + (1 - c)\frac{1}{1+e^{-a(\theta-b)}} \quad (1)$$

In IRT, test items are selected to yield the highest information content about the examinee by presenting items with difficulty parameter values that are closer to his ability value. This reduces time by asking fewer and relevant questions rather wider range ones while satisfying content considerations such as items or rules that are critical for a decision of access or scoring.

1) IRT parameter estimation

In order to determine the difficulty and discrimination parameters of a test item, IRT uses Bayesian estimates, maximum likelihood estimates or similar methods (MLE) [3, 4]. In the original IRT, an experiment is conducted to estimate these values for each item and at an assumed level of ability for various groups with associated values of IRT parameters using his judgment and experience. Nevertheless, by using our system we can also revise any initial values for these parameters. We model rule attributes as test items and rely on the policy administrator to provide the estimated probabilities.

2) IRT ability estimation

In IRT, responses to questions are dichotomously scored. That is, a correct answer gets a score of "1" and an incorrect answer gets a score of "0". The list of such results consist an item response vector. To estimate the examinee's ability, IRT utilizes maximum likelihood estimates (MLE) using an iterative process involving a priori value of the ability, the item parameters and the response vector as shown in (2). Here, $\hat{\theta}_s$ is the estimated ability within iteration s. a_i is the discrimination parameter of item i, where i=1,2,...,N. u_i is the response of the examine (1/0 for correct/incorrect). $P_i(\hat{\theta}_s)$ is the

probability of correct response from (1). $Q_i(\hat{\theta}_s)$ is the probability of incorrect response = 1- $P_i(\hat{\theta}_s)$ [3,4].

$$\hat{\theta}_{s+1} = \hat{\theta}_s + \frac{\sum_{i=1}^{N} -a_i\,[u_i - P_i(\hat{\theta}_s)]}{\sum_{i=1}^{N} a_i^2\; P_i(\hat{\theta}_s)\, Q_i(\hat{\theta}_s)} \qquad (2)$$

Then, the ability estimate is adjusted to improve the computed probabilities with the examinee's responses to items. This process is repeated until the MLE adjustment becomes small enough so that the change becomes negligible. IRT accommodates multiple stopping criteria such as: fixed number of questions, ability threshold or a standard error confidence level. The result is then considered an estimate of the examinee's ability and the estimation procedure stops. The ability or trait usually ranges from -∞ to +∞, but for computational reasons acceptable values are limited to the range [-3, +3].

C. Access Control and XACML

Access control policies specify which subjects may access which resources under some specified conditions [6]. An attribute-based access control policy specifies subjects, objects and resources using some attributes. XACML is an OASIS standard XML-based language for specifying access control policies [7]. In a typical XACML usage scenario, a subject that seeks access to a resource submits a query through an entity called a Policy Enforcement Point (PEP), which is responsible for controlling access to the resource. It forms a request in the XACML request language format and sends it to the a policy decision point (PDP), which in turn, evaluates the request and sends back one of the following responses: accept, reject, error, or unable to evaluate.

IV. USING IRT TO MANAGE AND CONTROL DIALOGUES FROM ONTOLOGIES

Fig. 3 shows the overall architecture of our system. We use derived or axiomatic facts of the ontology to create questions asked by our IVR system. Given that a large number of facts can be derived from our ontology, but only few questions can be asked during an interview, we use IRT to select the facts that are used to generate questions.

Our questions are automatically created without human involvement by combing English words or phrases such as "Does" or "Is-a" with ones chosen from the ontology of (subject, property, object) triples. The expectation is a dichotomous answer of either (yes, no) or (true, false). The ontological property names such as "is-a", "has-something" are prime candidates for creating true/false questions. Our system transforms the question

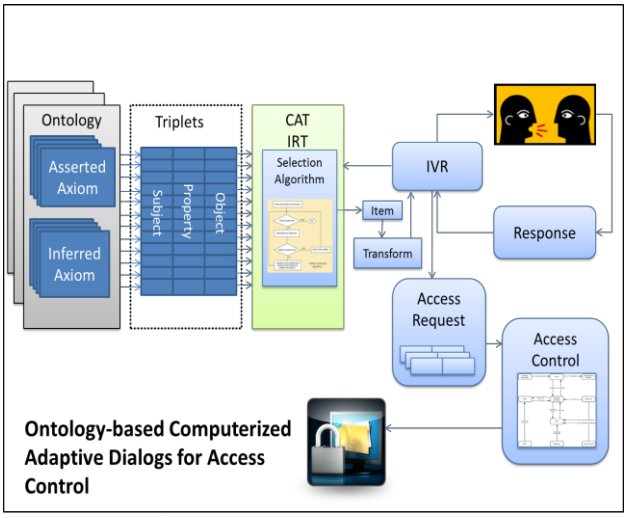


Fig. 3. Ontology-based IVR using IRT

into VoiceXML and plays to the user. Then the system waits for the user's utterance, and if the user provides one, the system's voice recognition software attempts to recognize the input and checks the correctness of the answer. Based on the answer, the IRT estimation procedure either increases a priori ability score or decreases it. The process continues until a predetermined level of ability or accuracy specified according to the application is reached.

Because ontologies produce a large number of facts, it would be impractical to run a dialogue that lasts hours in order to estimate user's ability. In our homeland security ontology uses 167 axioms. The reasoner was able to infer 94 facts raising the total number of axioms and candidate to generate questions to 273.

We use IRT to manage and control dialogue questions generated from a large pool of ontologically derived facts in a way that shortens the length of dialogues while keeping the maximum accuracy in estimating the user's trust. The IRT-based estimated (θ) represents the trust or confidence of the system in the person answering the questions in order to make an access decision.

We have used the OWL annotation property to assign IRT parameters to axioms. Annotations were selected in order to keep the semantics of the original ontology and structure intact. We annotate every asserted axiom in the ontology with IRT parameters, which are: difficulty (b), discrimination (a) and guessing (c). Currently, we assume all asserted axioms have the same default degree of difficulty and discrimination values of 1. The code snippet in Fig. 4 illustrates our annotation using Java with OWL API. An improvement to this approach would be to assign different values for difficulty and discrimination by using domain experts.

```
OWLAnnotationProperty irtDifficultyAP =
df.getOWL
AnnotationProperty(IRI.create("#irt_difficulty"
));
OWLAnnotation irtAnnotation =
df.getOWLAnnotation(
   irtDifficultyAP , df.getOWLLiteral(1.0));
for (OWLAxiom axiom : axioms) {
     OWLAxiom axiom2 = axiom.getAnnotatedAxiom
        (Collections.singleton(irtAnnotation));
     manager.addAxiom(ontology, axiom2);
}
```

Fig. 4. Java code for asserted axiom annotation

We weigh inferred facts more during the estimation process. We are calculating these parameter values from the number of explanation axioms used in each individually inferred fact. Our current scheme of difficulty value assignment is shown in Table II; where higher values or weights are assigned according to the number of explanation axioms used to infer a fact, and consequently the question generated from it is considered to be more difficult than one generated from an asserted fact. Fig. 5 illustrates a code snippet for inferred axiom annotation.

In our current work and for testing purposes we use a default value of "1.0" for discrimination and "0.0" for guessing, which practically neutralizes them leaving the difficulty parameter as the sole factor in estimating ability using equation 2. However, our solution and algorithm are based on the IRT two-parameter model, which relies on the item's difficulty and discrimination parameters. Fig. 6 shows our algorithm to estimate ability based on equation 2 [3]. Our system estimates the ability of a user after every answer to a question generated from an axiom before selecting and asking the next question. If the ability estimate exceeds the threshold then access is granted. If the threshold is not reached then additional questions are offered. If the estimated ability doesn't reach the threshold the dialog stops and access is denied. Depending on the application, the dialog might be run again giving a second chance. When the ability estimation again reaches a predefined threshold, the system concludes the dialog and conveys the decision.

TABLE II. IRT DIFFICULTY ASSIGNMENT BASED ON NUMBER OF AXIOMS IN EXPLANATION

Number of explanations	IRT Difficulty	
1	0	Easy
2-3	1	
4-5	1.5	Moderate
6-7	2	
8-9	2.5	
>=10	3	Hard

```
Set<OWLAxiom>
inferredAxioms=inferredOntology.getAxioms();
DefaultExplanationGenerator explanationGenerator
=new DefaultExplanationGenerator(
manager, factory, ontology, reasoner, new
SilentExplanationProgressMonitor());
for (OWLAxiom axiom : inferredAxioms) {
    Set<OWLAxiom> explanation =
explanationGenerator.getExplanation(axiom);
//Annotate inferred axioms using the number of
explanation
OWLAxiom tempAxiom =
axiom.getAnnotatedAxiom(Collections.singleton(irt
Annotation));
manager.addAxiom(inferredOntology, tempAxiom);
```

Fig. 5. Java code for inferred axiom annotation

The resultant decision is based on the IRT characteristics of the axiom and not on the number or the percentage of correctly answered questions as in traditional testing. The ability estimate produced by our implementation also comes with a standard error (SE) value that is a measure of the accuracy of the estimate. Equation (3) presents the formula used for standard error calculation [7].

$$SE(\hat{\theta}) = \frac{1}{\sqrt{\sum_{i=1}^{N} a_i^2 P(\hat{\theta}) Q(\hat{\theta})}} \quad (3)$$

Higher standard error indicates that the estimate is not very accurate, while lower values indicate higher confidence in the estimation. This too can be used as a means to discontinue the dialogue or use an alternate decision method.

V. IMPLEMENTING THE ONTOLOGY-BASED IVR SYSTEM FOR ENTRY CONTROL

Here, we present a prototype of our system showing the major components. It is not yet validated as a deployable system, but it works for the sample use case.

```
Algorithm 1: IRT Ability estimation
Input:a priori theta, Difficulty, Discrimination,
Answer
Output: posteriori theta, standard error
/* calculate theta and standard error*/
1:for (counter < items.length) do
2: itemDifficulty=parseFloat(difficultyArray[i]);
3:itemDiscrimination=parseFloat(discriminationArr
  ay[i]);
4:answer=parseFloat(answerArray[i]);
5:probTheta=calculateProbability(itemDiscriminati
  on,aTheta,itemDifficulty); // equation 1
6:thetaSplus1= claculateTheta(probTheta, thetaS);
  //equation 2
7:endfor;
8:estimatedTheta = thetaSplus1;
9:return thetaSplus1;
```

Fig. 6. Algorithm for ability estimation in IRT

1) *Voice Platform (Voxeo)*

We use the Voxeo's Prophecy local server as our voice platform for voice recognition and to run the dialogues. Java, Java Server Pages (JSP), and Java Script (JS) are used to implement the architecture modules and to implement IRT procedures used to estimates the user's ability/trust scores.

Voxeo's Prophecy is a comprehensive IVR and standards-based platform [15]. Some of the capabilities integrated into the platform are: automatic speech recognition, speech synthesis (Text-to-Speech), Software Implemented Phone (SIP) browser and libraries to create and deploy IVR or VoIP applications using VXML CCXML. It supports most of server side languages and has a built-in web server.

2) Item bank

In our work, we start with ontology, annotate every axiom with an "irt_difficulty" property of value "1". Then we use this ontology in the HermiT reasoner to infer implicit axioms and their explanations. The inferred facts are themselves annotated with "irt_difficulty" property and values calculated by factoring the number of explanation axioms using the schema stated in Table II.

For example, when annotating the inferred fact "*the friends of the Boston Attack Bomber*", which has an explanation that includes 11 axioms shown in Fig. 2, the *irt_difficulty* annotation would be "3.0"; which is the highest value on the scale of IRT difficulty parameter values in Table II. We assume that answering a question generated from a high-valued fact is a difficult task. Consequently, if the answer to a question derived from this fact is correct, the ability estimate would be impacted more positively than a correct, but easy one and more negatively if the opposite happens. An example is the asserted axiom that "Boston is located in Massachusetts". Because this is an asserted fact, it is annotated with value "1.0"; which makes a question generated from it an easy one and thus not affecting the ability estimate greatly.

This process is basically generating the item bank in CAT/IRT terminology. Each item in the item bank contains a question, an answer and IRT parameters. In addition to saving it as ontology in any of the supported formats, this item bank can also be supported by using a more specialized CAT/IRT platform like Cambridge University's Concerto [16].

3) *Generating dialogues from an ontology*

```
<form id="Begin">    <block>
<prompt bargein="true">
   Welcome to the United States. To accelerate
your entry, we will appreciate your responses to
some questions to verify your identity and
eligibility  </prompt>
<assign name="xacmlResource" expr="'point of
entry"/>
<goto next="#Resource"/></block>
</form>
```

Fig. 7. A sample Homeland security VoiceXML greeting form

The conversation starts with a menu in VoiceXML hosted on the local Voxeo Prophecy web server. The voice browser connects to the web server and converts text to speech and speech to text. Fig. 7 shows a sample VoiceXML code.

Fig. 8 shows our algorithm integrating ontology, IVR and IRT. This algorithm was successfully implemented using JavaScript and Java Server Pages (JSP) embedded in VoiceXML pages. The main steps are as follows:

- Load the ontology and parse the XML into Document Object Model (DOM).
- Extract the axiom's triplet (subject, property, object)
- Extract the axiom's IRT difficulty value from the annotation
- Establish a VoiceXML "For" loop that synthesizes a question from string or text values to speech (TTS). The question consists of an auxiliary verb, object, property and subject to test the correctness of an axiom.
- The system waits for a response. If there is one it converts it to text and recognizes it. If it adheres to grammar then a value is assigned as an answer.
- If there was no answer then VXML re-prompts the question up to a programmed number of times. If exceeded then an appropriate VXML is executed.
- The vector of binary answers is used to estimate the IRT ability.
- The loop continues until a threshold of θ or the maximum number of questions is reached.
- The IRT ability estimation algorithm, as illustrated in Fig. 6, takes the variables: answer vector, a priori θ, difficulty, discrimination and calculates a posteriori $\acute{\theta}$.
- If the answer is correct ("yes" or "true"), a value of "1" is assigned. If not, a "0" is assigned.
- The last posteriori $\acute{\theta}$ in the loop is the estimated user's ability θ and can be compared to a threshold value set by an administrator. Access is granted if ($\theta > threshold$) and denied otherwise.

```
Algorithm 2: dialogue access evaluation
Input: a priori theta, Difficulty,
Discrimination, Answer
Output: access control decision
/* make access control decision from
ontology*/
1:  domDocument=parse(ontology); // DOM
2:  subjectArray=getAxiomSubject(axiom);
3:  propertyArray=getAxiomProperty(axiom);
4:  objectArray=getAxiomObject(axiom);
5:  difficultyArray=getAxiomDifficulty(axiom);
6:  /*use voiceXML , JSP to generate dialog*/
7:  for (counter < items.length) do
8:     <vxml:Prompt> '[auxiliary verb]'
    +propertyArray[i] + " " + objectArray[i]
    +" "+ subjectArray[i];
9:     <vxml:Field>= user_utterance;
10:    response[i] =
    Field.voiceRecognition(user_utterance);
11:    if response[i]= 'Yes' or 'true'
12:       resultVector[i]=1;
13:    else
14:       resultVector[i]=0;
15: endfor;
16: theta = IRT_algorithm(resultVector,
    difficulty, discrimination,aPrioriTheta);
17: if theta > thetaThreshold
18:    permit;
19: else
20:    deny;
```

Fig. 8. Ontology-IVR algorithm with IRT

VI. EXPERIMENTAL RESULTS

Our implementation shows that efficient dialogs could be generated from ontologies that have been enhanced with IRT attributes. The successful implementation of the IRT in dialogues of questions and answers shortens the number of questions necessary to reach an accurate estimation of subject's ability, knowledge or trust by at least 50% as it has already been proved by the IRT literature [4, 5]. This reduction of the number of questions necessary to estimate the ability produces shorter dialogs without losing accuracy. Also, the use of IRT enables the use of multiple stopping criteria such as: fixed length number of questions or time, ability threshold and standard error confidence interval. The availability of large number of ontology axioms enables generating a set of questions different from another set to be generated immediately after the current user preserving privacy and protecting against question exposure, especially in voice systems. The success of dialog system depends upon multiple timing factors and scalability of supporting multiple users. Our on-going research addresses these two aspects.

VII. CONCLUSION

We have designed and implemented a novel IVR system that can dynamically generate efficient interactive voice dialogs from ontologies for entry control. We have used IRT to generate shorter dialogues between the system and a human speaker. IRT is useful in compensating for inaccurate voice recognition of answers during dialogs or accidental mistakes. Our entry control decisions are made based on an estimation of a level of trust in a subject derived from the importance or relevance of axioms in ontology. The use of IRT also enables the reordering of questions with the purpose of preserving privacy in IVR systems. With the advancement in the fields of mobile, cloud and cloud based voice recognition such systems become important in defence and physical security applications [17, 18, 19].

REFERENCES

[1] M. Ababneh, D. Wijesekera, J. B. Michael, "A Policy-based Dialogue System for Physical Access Control", The 7th STIDS 2012), Fairfax, VA, October 24-25, 2012.

[2] M. Ababneh, D. Wijesekera, "Dynamically Generating Policy Compliant Dialogues for Physical Access Control", CENTERIS 2013 - Conference on Enterprise Information Systems – aligning technology, organizations and people, Lisbon, Portugal. October 23-25, 2013.

[3] F. B. Baker, The basics of item response theory, ERIC Clearinghouse on Assessment and Evaluation, 2001.

[4] D. J. Weiss, G. G. Kingsbury, Application of computerized adaptive testing to educational problems, Journal of Educational Measurement, 21, 361-375, 1984,

[5] H. Wainer, Computerized Adaptive Testing: A Primer, Second Edition, Lawrence Erlbaum Associates Publishers, 2000

[6] M. Bishop, Computer Security: Art and Science, Addison Wesley, 2002.

[7] XACML, OASIS, URL: https://www.oasis-open.org/committees/tc_home.php?wgabbrev= xacml, accessed September 30, 2013.

[8] T. R. Gruber. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199-220, 1993.

[9] W3C, Web Ontology Language (Primer), http://www.w3.org/TR/owl2-primer/ , accessed August 22, 2013.

[10] W3C, SPARQL Protocol and RDF Query Language, URL: http://www.w3.org/2009/sparql/, accessed August 22, 2013.

[11] http://owlapi.sourceforge.net/reasoners.html, accessed August 22, 2013.

[12] http://hermit-reasoner.com/, accessed August 22, 2013.

[13] W3C Voice Browser Working Group, URL: http://www.w3.org/Voice, accessed August 22, 2013.

[14] W3C, Voice Extensible Markup Language (VoiceXML)(VXML), URL: http://www.w3.org/Voice/, accessed August. 22, 2013.

[15] Voxeo web site, URL: http://www.Voxeo.com, accessed August 22, 2013.

[16] Concerto IRT Platform, URL: http://www.psychometrics.cam.ac.uk/page/338/concerto-testing-platform, accessed August 22, 2013.

[17] Microsoft Windows Phone Speech, URL: http://www.windowsphone.com/en-us/how-to/wp7/basics/use-speech-on-my-phone, accessed September 3,2013.

[18] Apple Siri, URL: http://www.apple.com/ios/siri, accessed September 3, 2013.

[19] Google Android Mobile Search, URL: http://www.google.com/mobile/search/, accessed September 3, 2013.

Fast Semantic Attribute-Role-Based Access Control (ARBAC)

Leo Obrst[a], Dru McCandless[b], David Ferrell[a]
The MITRE Corporation
[a]McLean, VA
[b]Colorado Springs, CO
{lobrst, mccandless, ferrell}@mitre.org

***Abstract*—We report on our research effort, called Fast Semantic Attribute-Role-Based Access Control (ARBAC), to develop a semantic platform-independent framework enabling information originators and security administrators to specify access rights to information consistently and completely, in a social network environment, and then to rigorously enforce that specification. We use a modified ARBAC security model and an OWL ontology with additional rules in a logic programming and Java framework to express access policy, going beyond the limitations of previous attempts in this vein. We also experimented with knowledge compilation optimizing techniques that allow access policy constraint checking to be implemented in real-time, via a bit-vector encoding that can be used for rapid run-time reasoning.**

***Index Terms*—access control policy, attribute-based, role-based, Semantic Web, logic programming, knowledge compilation, social network, ontology, rule-based reasoning**

I. Introduction

This paper is a report of our effort to provide a semantic platform-independent framework so that information originators and security administrators can specify access rights to information consistently and completely, in a social network environment, and then to rigorously enforce that specification. In previous work [1], we discussed the architecture and some issues with optimization. In this paper, we introduce the architecture (adapted from [1]), but focus more on the optimization and implementation issues; as such, this paper can be viewed as a follow-on to [1].

For many sensitivity, privacy, and proprietary reasons, information sharing cannot be totally open. This is especially true for collaborative social environments such as the emerging MITRE Partnership Network (MPN), a large-scale environment for group-based (social network) information sharing among disparate governmental, commercial, academic, and other communities.

In addition, it is difficult to enforce unambiguous access rights and information privileges consistently and coherently and apply the access rules correctly and efficiently.

In a collaborative social environment, access control of information protecting privacy, security, and also enabling a complex range of policy respecting those requirements, is difficult.

To accomplish these objectives it is necessary to link a security policy model to a policy language with sufficient expressive power to ensure logical consistency. We used a modified Attribute-Role-Based Access Control (ARBAC) security model and an OWL ontology with additional rules in a logic programming framework to express access policy, going beyond the limitations of previous attempts in this vein, and then optimized with bit-vectors the runtime policy checking inference.

We focused on three aspects: expressivity, adaptability, and efficiency. We developed two implementations: one that transforms the policy model instance into a logic programming execution environment that includes rules; and a second that transforms the model instance into Java data structures, that in turn are optimized via a bit-encoding. In both cases, the prototype was embedded in a Java program that interfaces with external services, e.g., obtaining identity and access tokens (and their specific attribute information) from the authentication service.

The structure of the rest of the paper is as follows. In section II, we present the overall architecture and describe the runtime components. Then in section III, we briefly walk through the processing involved, followed in section IV by a discussion of the implementation. Section V addresses the optimization issues. We introduce related work in section VI, and finally, in section VII, we propose future directions.

II. System Architecture And Runtime Components

The general system architecture of the semantic ARBAC system is represented in Figure 1. It consists of three processes which flow from left to right. The three processes are: 1) the *Development* time process; 2) the *Transformation* time process; and 3) the *Execution* (runtime) process.

The Development process (the red rounded rectangle in Figure 1) involves:

1) The creation (or update) of the ARBAC ontology, represented in OWL and RDF, i.e., the semantic policy model (SPM); and

2) The instantiation of the specific ARBAC policy (policies) to be transformed and deployed, i.e., the semantic policy instance (SPI). This is an instance of the semantic policy model.

The Transformation process (the yellow rounded rectangle in Figure 1) involves developing and/or generating in Prolog and Java:

1) The transformer interpreter that will take the SPI and generate the runtime semantic policy instance (RSPI), which is the bit-vector representation of the policy + rules;
2) The attribute signature assignment engine (ASAE) which generates and updates the resource access registry (RAR);
3) The RAR, which captures the attributes of the resources in bit-vector representation, indexed by resource URI;
4) The runtime user access routine (RUAR);
5) The runtime inference engine (RTIE) which will execute the RSPI using the RUAR.

The Transformation process can thus be considered a knowledge compilation process, where source semantic models and their interpreting engines get transformed to efficient Execution time process objects.

The Execution process (the blue rounded rectangle in Figure 1) thus includes the RAR, ASAE, RTIE, and the RUAR, in addition to access to the Development and Transformation models and data.

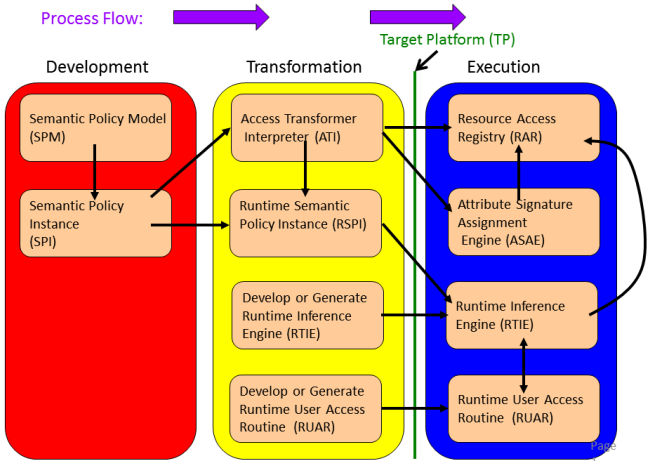


Fig. 1. Fast Semantic ARBAC System Architecture

Figure 2 displays the runtime system components of the Fast Semantic ARBAC system. The runtime system components view represents most components of the system architecture modules displayed in Figure 1, but focuses on their relationships at runtime only.

A. Semantic Policy Model (SPM)

The SPM consists of the OWL ontology classes, object properties, and data properties. The major classes consist of: *Subject* (the person, organization, software that requests specific access to a resource), *Action* (the kind of access requested, e.g., read, write, create, delete, execute, etc.), *Resource* (the object needing to be accessed by a subject: executable, graphic, text, sound, video, hardware, etc.), *Environment* (salient aspects of the space or session's environment, e.g., risk or alert level, entry network domain), *Role* (traditional roles such as administrator, expert, end user, developer, etc., that are also related to groups), and related notions: *Authentication* (how one authenticates one's identity and so, derivatively, one's potential access rights), *Security* (can span information security notions such as protocols, standards, user- and group-level passwords, encryption methods, hashing algorithms and values, etc.), *Classification Level* (proprietary, sensitive, confidential, secret, top-secret, etc.), *Identity* (Public Key Infrastructure [PKI], digital certificates, etc.), *Time* (time-stamps, time intervals with respect to various policy notions), etc.

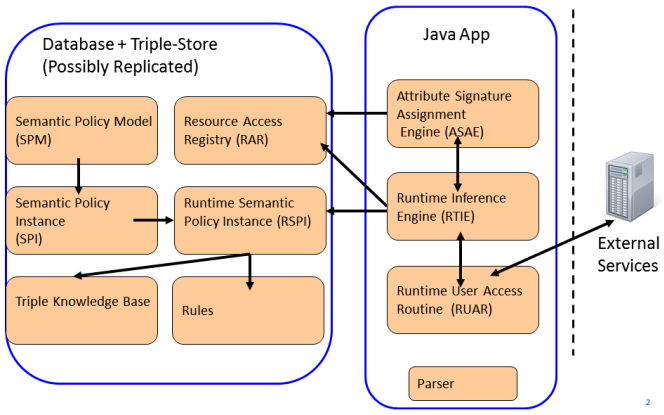


Fig. 2. ARBAC Runtime System Components

In addition, rules are a very important component of the semantic policy model (SPM). Rules exist outside of the OWL ontology per se, but are based on the classes and properties specified in the ontology. Rules were expressed initially in Prolog, and then in Java code for the second prototype. Rules are potentially recursive and express logical constraints among and across class and property values (instances). Some examples are given below.

The SPM represents a set of generic semantic components for ARBAC policy, and thus constitutes a family of potential specific ARBAC instantiations.

B. Other Components of the Architecture

For more detailed descriptions of other components of the architecture, including the SPI, RSPI, RAR, ASAE, RIE, RUAR, the OWL parser, and external service interface, we direct interested readers to [1].

III. Access Decision Process Flow and Walkthrough

The following depicts the access decision process flow.

- Initially, the Policy/Rules KB is read and loaded (including any general rules that apply to all circumstances) by the inference engine.
- Then a request comes in containing the Subject, Resource, Action, and Environment.
- The Subject's Group membership is looked up and formed.
- An initial Resource/Group/Access check may be performed.
- For some common accesses these may be cached, or may require no further processing if a quick decision can be made.
- Otherwise, the appropriate rule set is generated and populated with: any referenced access rule (pre-filtered to keep the KB small and fast), all facts about the Subject, Resource, Groups, and Environment, and General (generally applicable) rules.

- The rule set is passed to a runtime inference engine which evaluates the truth of the permission statement (something along the lines of allow(Subject, Access, Resource)).
- The Inference Engine passes back the permission decision.

The semantic policy model (SPM) is the holder of much of the underlying knowledge. Its contents include:

- Ontology
- Access Rules
- Group Membership Rules
- General Rules

The Access Rules ultimately determine whether an action can be performed on a resource (a 'Privilege' to denote the pairing of actions and resources); each rule has three parts:

1. The head, or consequence, which is always a privilege (e.g., hasPrivilege(subject22, read,medicalRecord66)). This leaves the body of the rule which for convenience is broken into 2 parts:
2. The Group membership required to obtain the privilege, and
3. Any additional requirements, expressed in terms of environment variables.

Example:

hasPrivilege(Subject, Action, Resource)

← agent(Subject), member(Subject, Group), environmentalConstraints(Group, Action, Resource, Environment), groupWithPrivilege(Group, Action, Resource, Environment).

Premises:

- All access decisions can be expressed as a *privilege* ← *requirements* rule.
- All role or subject attributes can be expressed as group membership.
- Group membership is both dynamic and contextual.
- Resources and their attributes are known a priori. If resources and attributes can change arbitrarily dynamically, this will decrease performance.

Knowledge of four things is used to resolve a permission question:

1. The Subject (the entity requesting the permission)
2. The Resource that the Subject is requesting permission about
3. The Action that the Subject wishes to perform
4. The Environment, which is a set of facts/assertions that the rules may take into account in order to make a permission determination.

The result will be either a yes or no answer as to whether permission is granted.

The access rules can have fairly complicated group membership conditions (e.g., a doctor who is an associate of a patient's primary care physician can have read access to that patient's medical record). Therefore, determining group membership may rely on a number of General Rules to help resolve the inferences (e.g., a doctor may be a member of a group; if another doctor is also a member of that group, then that doctor is an associate of the first doctor, etc.). By making group membership dynamic we can keep the access rules general.

IV. IMPLEMENTATION

The Fast Semantic ARBAC software prototype was designed to show how a system could quickly make access decisions based on the attribute values of the requesting agent. How the agent obtained the attribute values is outside the scope of the prototype; the ARBAC system is provided these from a separate source, projected to be a session authentication token (with a prescribed lifespan), that points to the attribute store, which has been obtained and encoded by the ARBAC system.

To achieve this, five conceptual classes were defined that constitute the "ARBAC view" of the world: Agents, Resources, Groups, ResourceCollections, and Policies. Two of these are collections, or sets: Groups (collections of Agents) and ResourceCollections (collections of Resources). They are hierarchical, e.g., one group may be a subset of another group, so any member of the subset group is automatically a member of the larger group. The other three classes are "flat" in an ontological sense, but contain many instances. Agents have (at least) a unique ID, and zero or more attribute/value pairs, which contain values that may be assigned to them by an organization or may be values contained in a security token. A Group is a set of Agents; group membership can be expressed in two ways: directly (an Agent by his/her ID value is asserted to be a member of a specific group) or indirectly (by specifying a set of attribute/value pairs an agent must possess in order to be a member of that group; any agent having all of the specified attribute/value pairs is considered a member of the group). Each group also has a unique ID. Unique IDs are considered special attributes and are assigned by the attribute signature assignment engine (ASAE), which updates the resource access registry (RAR). Agent IDs in the future will probably inherit the IDs of the identity token received from the external authentication service.

Resources and ResourceCollections are organized similarly to Agents and Groups. Resources also have a unique ID assigned by the attribute signature assignment engine (ASAE), and possess attribute/value pairs (such as ownedBy:: someOrganization, or locatedAt:: area). ResourceCollections likewise are sets of Resources, and membership can also be asserted directly or indirectly using a set of attribute/value pairs that a Resource must have.

Policies are different from the other four classes, in that they specify the "access rules" of what it takes for an Agent to perform some action on a Resource. In essence, a policy is just a 3-tuple containing a reference to a ResourceCollection ID that the policy controls, a reference to the Group ID to which an Agent must belong, and the action (from an enumerated set) which the Agent is requesting to perform.

The result is a simple but very flexible way to organize authorization decisions about accessing resources. In addition to general group membership, some special cases are also supported. For instance, a ResourceCollection can be created

to contain a single resource in order to directly control it. Similarly, a Group can be defined to consist of a single agent thus allowing individualized policies. Again, Groups and ResourceCollections may be organized in a hierarchy which simplifies policy creation and application. Some advanced access control mechanisms, such as an expiration date/time for an agent's token value, or the ability to specify negative conditions (e.g., agents which have a certain attribute/value pair(s) are NOT allowed access) are not implemented in this prototype, but are not precluded by this approach (i.e., they could be added at a later date without having to re-design the prototype system).

The ARBAC software is able to make quick authorization decisions because 1) most of the required information is known a priori and 2) the actual decision becomes a largely lookup-and-compare operation. The policies and resource attributes are known and stored in a location accessible to the ARBAC system. The Group and ResourceCollection definition rules are also known ahead of time and stored (although these may need to be recomputed from time to time). The agent's attribute/value pairs are passed to the ARBAC system (usually via a secureID token, but it can be done in other ways) once the agent logs onto the system. The Groups to which the Agent belongs can then be pre-computed right after login (before the Agent even selects a Resource, in most cases). Once the agent selects a Resource and the action he/she wants to take, a series of lookups take place. First, all of the policies related to the Groups to which the Agent belongs and allow the requested Action are obtained. Next, all of the IDs of the ResourceCollections to which the Resource belongs are obtained. Then the retrieved policies are examined to see if any of them contain a reference to any of the relevant ResourceCollections. If any one of them does, then that allows the Agent to access the requested Resource and perform the desired action. If none of the policies contains a reference to any of the possible ResourceCollections, then the action is not allowed.

The actual implementation of the system allows for several possibilities. Based on our work in FY12, the initial design represented each of the five conceptual classes as OWL classes, and each instance as an OWL individual. Attribute/value pairs were implemented as OWL datatype properties, as were the policy tuples. While some of the reasoning (such as class hierarchy subsumption) could be done in OWL, most of the actual policy/rule reasoning was done using Prolog. The ARBAC system converted the (hierarchically extended) information into Prolog assertions and then made a prolog query to see if a particular Agent/Resource/Action combination was allowable. While this proved workable, expressing all of the information in OWL (and using the Jena OWL reasoner to do some of the pre-computation) turned out to be somewhat cumbersome. Furthermore, the OWL format is not very interoperable with what are likely to be the other components of a true ARBAC system (such as other databases). Since only a small portion of the OWL semantics were needed, it was decided to generalize the expression of the ARBAC data by allowing it to be held in other formats, e.g., JSON (Java Script Object Notation).

Using JSON instead of OWL (with Jena) resulted in a performance increase. Also, because many data sources support JSON this approach will make interoperability much easier. Another implementation change was to use a direct bit vector approach in Java for policy evaluation, rather than Prolog. The idea is that by keeping everything in Java (Prolog requires a call to an external .dll or .so application) and using the inherent efficiency of bit reasoning, performance would increase further. So a parallel implementation using the standard Java BitSet class was created, whereby each attribute/value pair is assigned a bit position at runtime. Group membership and ResourceCollection membership were then pre-computed using a set of bits (i.e., a bit vector). When an agent selects a Resource, all of the Policies are retrieved based on the pre-computed ResourceCollections, and these are compared with the set of the Agent's Groups. If any Group is found in any of the policies, then the action is approved. Given the small set of data available, it was not possible to determine which approach (Prolog based or bit vector based, or both) will have the better performance at scale; this determination will need to be made during a follow-on test and integration effort.

V. Optimization: Bit-Encoding

Bit representation for ontology constructs (classes, properties, etc.), subsumption, and rule reasoning must address two related notions:

1) Efficiency of the representation in space and time. This includes efficiency of the encoding for storage purposes, but also compaction/compression techniques. It also includes the time required to perform the offline, development time encoding, as well as the time required to do the matching, subsumption computations, and automated reasoning performed at runtime.
2) Incremental encoding, i.e., making modifications dynamically during runtime to ontology constructs and rules, potentially recomputing the encodings of ontology constructs and rules, and then continuing efficient reasoning.

A. Ontology Constructs

The primary ontology constructs we use are the following:

- *Group:* A subclass of Collection. There are Classes of Groups (such as the Federally Funded Research and Development Center [FFRDC] class) and there are instances of Classes that are groups (e.g., the instances of the FFRDC class, such as MITRE, Aerospace, Los Alamos National Lab, etc.)
- *Resource:* A resource is any hardware, software, or service.
- *ResourceCollection:* A subclass of Collection. There are Classes of ResourceCollections and there instances of Classes that are resource collections.

- *User:* A user (agent) is generally a person, but could be a software agent.
- *Policy:* A policy is a set of access constraints on a Group or Resource created by a User who has the requisite permissions to create the policy.
- *Access:* The kind of access a User has to a Resource, as permitted by a Policy. Examples: Create, Read, Write, Delete, Execute, etc.

Because we are focusing primarily on "attributes" for access control, whether or not a User U belongs to a specific Group is a Boolean attribute, with value either 'true' or 'false' (of value 'true' if the User U is a member of a Group G, else of value 'false'). Similarly, whether or not a Resource R is a member of a ResourceCollection RG is a Boolean attribute. If it helps us in our processing, even a User U can be considered a singleton Group, i.e., a specific instance of a Group having just one member, U.

We assume a User U can create a Policy P (perhaps of a specific type) that grants another User U' specific Accesses A to a Resource R of ResourceCollection RC if the User is a member of some Group G and Group G 'owns' the ResourceCollection. Other policies may specify Roles, etc., which we are not yet addressing here.

The bit-representation for Group (and Resource) constructs is similar to the following, naïve representation:

Table 1. User Groups: Bit Representation

	G1	G2	G3	G4	G5	G6	G7	G8	G9
U1	1	1	0	0	0	0	0	0	0
U2	0	1	1	0	0	0	0	0	0
U3	0	0	1	1	1	0	0	0	0
U4	1	0	1	1	1	1	0	0	0

B. Subsumption

Subsumption is the relatively simple automated reasoning that can be done on hierarchies of classes, i.e., the taxonomic subclass 'backbone' of the ontology. These subclass hierarchies are important for ontologies, but also important for strongly typed programming languages, which perform subsumption reasoning as 'type inference' over the formal types of constructions in the specific program.

Ait-Kaci et al [4] proposed a number of bit-representations that could be used for very efficient subsumption reasoning, by plungeing the hierarchy of classes (or types), which typically constitutes a 'partially ordered set' (poset), into a boolean lattice, thus enabling efficient Greatest Lower Bound (GLB) and Least Upper Bound (LUB) operations, and efficient transitive closure. In an arbitrary poset, neither the GLB or the LUB is guaranteed to exist, but there are formal structural embeddings one can perform on the poset into an order-preserving structure, a semilattice, a lower semilattice in this initial case, which preserves the GLB, sometimes called a meet-semilattice, which says that for any nonempty finite subset of poset, there is a GLB. Note that the *ordering relation* on the elements of the poset (which define the poset) is typically notated as $\leq$, e.g., a $\leq$ b, where $\leq$ is reflexive, antisymmetric, and transitive.

An ontology *subclass* relation is an ordering relation on the classes, i.e., reflexive, antisymmetric, and transitive. OWL provides a *top* (greatest or most general) and *bottom* (least or most specific) class, called respectively Thing and Nothing, which makes OWL into a language able to model *bounded (semi-) lattices*. Bottom is often notated as $\perp$, with top notated as $\top$.

C. Encoding Bit Representations of Subsumption and Inheritance

We will discuss encodings proposed in the literature, beginning first with a naïve bit matrix representation. For all of these encodings, we adapt the example used by [17, p. 16-17], displayed in graph form as the ontology of classes in Figure 3 (where the isa relation is taken to be synonymous with the subclass relation). We use this example, rather than one drawn from our domain ontology, simply because our ontology does not currently have much depth and no multiple inheritance, which this example has. Note that these 'role' subclasses are not ontologically correct, but have been accommodated to a simple example.

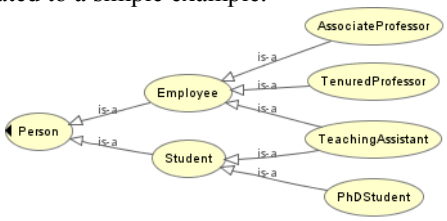


Fig. 3. Academic Role Ontology

Table 2 displays the naïve bit matrix representation for this ontology's subsumption relations. Note that the bit assignment goes as follows:

1) Initially assign 1 (true) for every class (i, j) (where i is the row, j is the column) and itself, because every class subsumes itself. This means there is a diagonal with value 1 from (1, 1) to (n, n).
2) Then for each cell of the matrix (i, j), if the class i is an ancestor of class j, assign the value 1, otherwise assign the value 0.

Table 2. Naïve bit matrix representation of Subsumption

i: row j: column	**Person**	**Student**	**Employee**	**Associate Professor**	**Tenured Professor**	**PhD Student**	**Teaching Assistant**
Person	1	1	1	1	1	1	1
Student	0	1	0	0	0	1	1
Employee	0	0	1	1	1	0	1
Associate Professor	0	0	0	1	0	0	0
Tenured Professor	0	0	0	0	1	0	0

PhD Student	0	0	0	0	0	1	0
Teaching Assistant	0	0	0	0	0	0	1
⊥	0	0	0	0	0	0	0

This encoding thus is the reflexive, transitive closure of the (antisymmetric) subclass (isa) hierarchy of Figure 4.

The naïve bit-assignment algorithm as represented in Table 2 is bottom-up, with an implicit 'bottom' (⊥). The classes Employee and Student, and then Person, are the only classes which have subclasses.

Subsumption between two classes can then be computed in constant time using a binary AND operation on the bit vectors of the two classes. The subsumption operator over the bit-encoded classes is defined as follows.

Definition: Subsumption over Bit-Encoded Classes:

Let $x_1, \ldots, x_n$ be classes in a subclass hierarchy, γ be an bit-encoding function, and $\sqsubseteq$ be the *subsume* relation (where α, β are classes and $\alpha \sqsubseteq \beta$ is read as 'class α subsumes class β'):

Then the following holds:

i. $\gamma(x_i) \sqsubseteq \gamma(x_j) \leftrightarrow \gamma(x_i)$ AND $\gamma(x_j) = \gamma(x_j)$

[the encoding of the first class subsumes the encoding of the second class if and only if the binary AND of those encodings is equal to the encoding of the second class]

ii. $\gamma(x_j) \not\sqsubseteq \gamma(x_i) \leftrightarrow \gamma(x_j)$ AND $\gamma(x_i) \neq \gamma(x_j)$

[the encoding of the first class does not subsume the encoding of the second class if and only if the binary AND of those encodings is not equal to the encoding of the second class]

Example 1: Does TeachingAssistant subsume AssociateProfessor?

I.e., does AssociateProfessor occur in the transitive closure of the subclass relation of TeachingAssistant?

SubsumeS (TeachingAssistant, AssociateProfessor)

= AND (0000001, 0001000) = 00000000, i.e., no.

Example 2: Does Person subsume TeachingAssistant?

Subsumes (Person, TeachingAssistant)

= AND (1111111, 0000001) = 0000001, i.e., yes, because the result 0000001 = 0000001 (the encoding for TeachingAssistant.

Example 3: Does Employee subsume Student?

Subsumes (Employee, Student)

= AND (0011101, 0100011) = 0000001, i.e., no, because the result 0000001 ≠ 0100011 (the encoding for Student).

What if one wants at runtime to add a new class incrementally (dynamically) after the above bit-representation has been generated at development time? We add the new class ResearchAssistant to the original ontology, resulting in Figure 4.

Recomputing our bit-matrix, we arrive at the following, Table 3. Note that we have to add a new bit by creating a new row and new column for ResearchAssistant, which we add as a new i+1 row and a new j+1 column into the matrix (but above the implicit □).

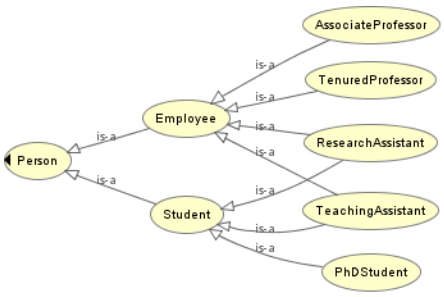


Fig. 4. Academic Role Ontology + ResearchAssistant

If we added the new bit as a new row and new column at the beginning of the matrix, then we would maintain the 1-bit diagonal we saw in Table 2. In addition, of course, we have to update the entries in the new Research Assistant column with their values (1 if an ancestor of Research Assistant, 0 otherwise). The naïve bit-encoding of Subsumption requires n2 bits.

Table 3. Naïve bit matrix representation of Subsumption with Incrementally Added ResearchAssistant Class

i: row j: column	**Research Assistant**	**Person**	**Student**	**Employee**	**Associate Professor**	**Tenured Professor**	**PhD Student**	**Teaching Assistant**
Person	1	1	1	1	1	1	1	1
Student	1	0	1	0	0	0	1	1
Employee	1	0	0	1	1	1	0	1
Associate Professor	0	0	0	0	1	0	0	0
Tenured Professor	0	0	0	0	0	1	0	0
PhD Student	0	0	0	0	0	0	1	0
Teaching Assistant	0	0	0	0	0	0	0	1
Research Assistant	1	0	0	0	0	0	0	0
⊥	0	0	0	0	0	0	0	0

Ait-Kaci et al [4] propose a number of new methods for encoding subsumption. Their first method requires a bottom-up (from the terminal classes to the root class) computing of the binary OR of the bits assigned to children classes, the result of which becomes the bit-encoding of their parent classes. New bits are introduced whenever a parent has just one class and

whenever a false positive subsumption would result. If incremental updates to the encoding are necessary, there are potential complications. If one wants to add new leaf (terminal) class nodes to the hierarchy, such as we did with ResearchAssistant above, there are no issues. However, if one wants to add new non-terminal (or root) nodes, there are complications. If a class C_j is added that has the same inheriting subclasses as an existing class C_i, then a new bit must be added to re-encode the existing class and all of its ancestors too. In addition, any new non-terminal class will have to have the ancestors of its children classes checked for conflicting encodings.

For a discussion of other bit-encoding techniques, the interested reader is directed to [17, pp. 16-23]. There are other encoding approaches, including interval-encodings. Interval-based encodings compute non-overlapping codes for the children within the interval of the parent, but do not support multiple inheritance.

In fact, although each of the above approaches out-perform the naïve encoding, all of them have some issues (except perhaps [17], which relies on binary representation of prime numbers) with incremental (dynamic) updates, requiring some recomputation of encodings and determination of conflicts, which in turn may require recomputation of encodings.

Rules too may be given encodings, but space limitations preclude a discussion of this topic here, but see [8] for Boolean satisfiability (SAT) reasoning using bit-matrices.

VI. Related Work

There is much previous related research across multiple dimensions (access control regimes, policy languages and approaches, specialized languages (and logics) vs. ontology approaches, knowledge compilation issues, bit-vector and other optimization approaches, social network approaches, privacy vs. security issues and approaches, etc.) that have influenced our current and impending work.

In order to accomplish our objectives it was necessary to link a security policy model to a policy language with sufficient expressive power to ensure logical consistency. We extend the NIST Role-Based Access Control (RBAC) security model [15] and related approaches [18-19], as have many other researchers to include attributes, and extend the Web Ontology Language (OWL) with additional rules to express access policy using logic programming, and beyond the limitations of [20]. Unfortunately, given our own space limitations here, we cannot do an extensive comparison of our approach across the multiples dimensions with other approaches, nor justly describe those other approaches.

In addition, there is extensive research in more general policy-based approaches that could be employed also for access control [21-22].

There are other Semantic Web-based approaches (including [22]), some of which address more specifically social network types of applications [23, 24].

For implementation in real-time, via a bit-vector or other efficient encodings that can be used for rapid run-time reasoning, we've looked at [2-6, 7-12, 17]. For bit-vector representation to support RDF triples, we investigated [11-14].

Our own previous work addressed issues in translating OWL/RDF ontologies and Semantic Web Rule Language Rules (SWRL) [25] into logic programming for efficient runtime reasoning, and employing knowledge compilation techniques [26-28], which we also generalized to address services using first-order logic theorem provers and for ontology alignment [29].

VII. Future Work

Although we have investigated and implemented some optimizations, e.g., extensionalization and delayed rule evaluation, we have only rudimentarily implemented the second-level of optimization we intended, i.e., the bit-representation execution at runtime.

If we had additional time, we intended to implement the prime-number bit-encoding of subsumption described in [17]. In general, for the restricted reasoning we need for access control policy enforcement as described in this paper, and given the probable volume of access request determinations (and thus subsumption and equivalence checks, rule execution) we foresee needing in a complex collaborative social network environment such as the MPN, optimized efficient automated reasoning is necessary. Traditional, more general description logic reasoners were deemed too slow (Pellet, etc.) In addition, most proposed bitmap encodings for subsumption and type reasoning are efficiently statically initialized and then used, but dynamically updating the subsumption/type hierarchy, i.e., adding, deleting, modifying classes and properties (which will happen, under the Open World Assumption of OWL and first-order logic), leads to degraded performance and increasingly baroque re-encodings to avoid conflicts.

Therefore, we would consider implementing the bit-encoding scheme based on assigning prime numbers to nodes in the class and property subsumption graphs, as developed by Preuveneers and Berbers [17, 30]. Adding a new class or property does not require re-encoding. Furthermore, the encoding automatically provides us the direction of the relationship. Modular hierarchies, each separately encoded, with very efficient subsumption-checking, are the result. Figure 5 depicts a subclass hierarchy encoded using prime numbers.

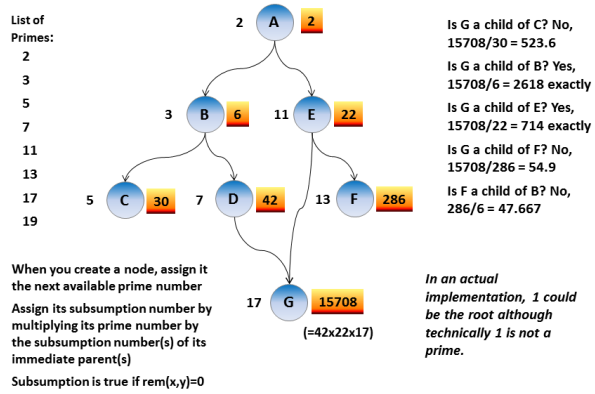


Fig. 5. Prime Number Encoding for Class Subsumption

In addition to the use of prime numbers, the scheme of [17, 30] defines a compact binary matrix representation of the inheritance relationships, which we will not go into here.

Evaluation done in [30, p. 32] shows that subsumption testing in his scheme is much faster than that of some major existing description logic reasoners, on the order of 250 times faster than Pellet. An evaluation performed on a different project we are involved in, written in C/C++ demonstrated 1000% improvement using this method of subsumption checking over the previous naïve, breadth-first search of the subsumption graph.

ACKNOWLEDGMENT

REFERENCES

[1] Obrst, L.; D. McCandless; D. Ferrell. 2012. "Fast Semantic Attribute-Role-Based Access Control (ARBAC) in a Collaborative Environment." The 7th IEEE International Workshop on Trusted Collaboration (TrustCol 2012), October 14–17, 2012, Pittsburgh, PA.

[2] Abadi, D. J.; A. Marcus; S. Madden; K. J. Hollenbach. 2007. "Scalable Semantic Web Data Management Using Vertical Partitioning." In Proceedings of VLDB, pages 411~422, September 2007.

[3] Ait-Kaci, H. 1984. "A Lattice-Theoretic Approach to Computation Based on a Calculus of Partially-Ordered Type Structures." Ph.D thesis, Computer and Information Science Dept., Univ. of Pennsylvania, Philadelphia, PA.

[4] Ait-Kaci, H.; R. Boyer; P. Lincoln; R. Nasr. 1989. "Efficient Implementation of Lattice Operations." TOPLAS 11-1-1989.

[5] Blandford, D. K.; Blelloch, G. E.; and Kash, I. A. 2003. "Compact representations of separable graphs." Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (Baltimore, Maryland, January 12 - 14, 2003).

[6] Blandford, D. K.; Blelloch, G. E.; and Kash, I. A. 2004. "An Experimental Analysis of a Compact Graph Representation." In Proceedings of ALENEX04.

[7] Caseau, Y.; M. Habib; L. Nourine; O. Raynaud. 1999. "Encoding of multiple inheritance hierarchies and partial orders." Computational Intelligence 15 (1), 50-62.

[8] Dershowitz, N. 2008. "Bit Inference." Workshop on Practical Aspects of Automated Reasoning, August, 2008, Sydney. 26-35.

[9] Fall, A. 1995. "Heterogeneous Encoding." In Proceedings of International KRUSE Symposium: Knowledge Retrieval, Use, and Storage for Efficiency, Gerard Ellis, Robert Levinson, Andrew Fall, Veronica Dahl, eds., Santa Cruz, CA, Aug. 11-13, pp. 134-146 (1995).

[10] Krall, A.; Vitek, J., Horspool, 1997. "Near optimal hierarchical encoding of types." 11th European Conference on Object Oriented Programming (ECOOP'97). Springer (1997).

[11] McGlothlin, J. P.; L. Khan, B. Thuraisingham. 2011. "RDFKB: A Semantic Web Knowledge Base." IJCAI, 2011.

[12] McGlothlin, J. P.; L. Khan. 2008. "RDFVector: A Scalable Data Model for Efficient Querying of RDF Datasets." http:// www.utdallas.edu/~jpm083000/ssDBM.pdf.

[13] McGlothlin, J.P.; L. Khan. 2010b. "Efficient RDF data management including provenance and uncertainty." IDEAS, 193-198, August 2010.

[14] McGlothlin, J. 2010. "RDFVector: An Efficient and Scalable Schema for Semantic Web Knowledge Bases." PhD Symposium, 7th Extended Semantic Web Conference (ESWC 2010), Heraklion, Greece. May 30 – June 3, 2010..

[15] http://csrc.nist.gov/groups/SNS/rbac/.

[16] Neumann, T.; G. Weikum. 2009. "RDF-3X: a RISC-style engine for RDF." In Proc. of VLDB, pages 647-659, September 2009.

[17] Preuveneers, D.; Berbers, Y., 2006. "Prime numbers considered useful: Ontology encoding for efficient subsumption testing," Tech. Rep. CW464. http://www.cs.kuleuven.be/publicaties/rapporten/cw/CW464. Department of Computer Science, Katholieke Universiteit Leuven, Belgium (October 2006).

[18] Sandhu, R. 1998. "Role-based access control." In M. Zerkowitz, editor, Advances in Computers, volume 48. Academic Press.

[19] Sandhu, R.; E. J. Coyne; H. L. Feinstein; and C. E. Youman. "Role-based access control models." 1996. IEEE Computer, 29(2):38–47, February 1996.

[20] Finin, T.; A. Joshi; L. Kagal; J. Niu; R. Sandhu, W. Winsborough; and B. Thuraisingham. 2008. "ROWLBAC: representing role based access control in OWL." In Proceedings of the 13th ACM symposium on Access control models and technologies (SACMAT '08). ACM, New York, NY, USA, 73-82.

[21] Tonti, G.; J. M. Bradshaw; R. Jeffers, R. Montanar; N. Suri; and A. Uszok. 2003. "Semantic web languages for policy representation and reasoning: A comparison of kaos, rei, and ponder." 2nd International Semantic Web Conference (ISWC2003). Springer-Verlag.

[22] Uszok, A.; J.M. Bradshaw; J. Lott; M. Breedy; L. Bunch; P. Feltovich; M. Johnson; H. Jung. 2008. New Developments in Ontology-Based Policy Management: Increasing the Practicality and Comprehensiveness of KAoS, IEEE Workshop on Policies for Distributed Systems and Networks, 145-152.

[23] Carminati, B.; E. Ferrari; and A. Perego, "Rule-based access control for social networks," in Proc. OTM 2006 Workshops, ser. LNCS, vol. 4278. Springer, Oct 2006, pp. 1734–1744.

[24] Masoumzadeh, Amirreza; James Joshi. 2010. "OSNAC: An Ontology-Based Access Control Model for Social Networking Systems." Social Computing (SocialCom), 2010 IEEE Second International Conference on Social Computing, 20-22 Aug. 2010, Minneapolis, MN, 751 – 759.

[25] Horrocks I.; Patel-Schneider, P.; Boley H.; Tabet, S.; Grosof, B.; Dean, M. 2004. "SWRL: A Semantic Web Rule Language Combining OWL and RuleML." /www.w3.org/Submission/SWRL/ .

[26] Samuel, K.; L. Obrst; S. Stoutenberg; K. Fox; P. Franklin; A. Johnson; K. Laskey; D. Nichols; S. Lopez; and J. Peterson. 2008. "Applying Prolog to Semantic Web Ontologies & Rules: Moving Toward Description Logic Programs." Journal of the Theory and Practice of Logic Programming (TPLP), M. Marchiori, ed., Cambridge University Press, Volume 8, Issue 03, May 2008, 301-322.

[27] Samuel, K.; L. Obrst. 2007. "Answer Set Programming: Final Report on a Comparison Between ASP and Prolog for Semantic Web Ontology and Rule Reasoning." October, 2007. MITRE MTR090069.

[28] Obrst, L; Stoutenburg, S; D. McCandless; D. Nichols; P. Franklin; M. Prausa; R. Sward. "Ontologies for Rapid Integration of Heterogeneous Data for Command, Control, & Intelligence." Chapter in: Obrst, Leo; Terry Janssen; Werner Ceusters, eds., 2010. Ontologies and Semantic Technologies for the Intelligence Community. Amsterdam, The Netherlands: IOS Press.

[29] McCandless, Dru; Leo Obrst. 2009. "Dynamic Web Service Chaining using OWL and a Theorem Prover." 3rd IEEE International Conference on Semantic Computing, Berkeley, CA, USA - September 14-16, 2009.

[30] Preuveneers, D.; Y. Berbers. 2008. "Encoding Semantic Awareness in Resource-Constrained Devices," IEE Intelligent Systems, March – April, 2008.

Supporting Evacuation Missions with Ontology-Based SPARQL Federation

Audun Stolpe, Jonas Halvorsen and Bjørn Jervell Hansen
Norwegian Defence Research Establishment (FFI)
P O Box 25
2027 Kjeller, Norway
Email: {audun.stolpe | jonas.halvorsen | bjorn-jervell.hansen}@ffi.no

***Abstract*—We study ontology-based SPARQL federation in support of coordinated action by deployed units in military operations. It is presumed that bandwidth is limited and unstable. Thus, we need an approach that generates few HTTP requests. Existing techniques employ join-order heuristics that may cause requests to multiply as a factor of the number of joins in a query. This can easily lead to an amount of traffic that exceeds network capacity. We propose an approach that builds an in-memory excerpt of the remote sources, sending one request to each source. A query is answered against this excerpt, which is a provably sound and complete representation of the sources wrt. query answering. The paper ends with a case study involving three military sources used for planning evacuation missions.**

I. Introduction

The planning of evacuation missions is a complex and important process in military operations. One of the most challenging aspects, is making all necessary information available to the decision makers. These information fragments will typically be distributed across different systems.

This is particularly the case when the military force conducts its operations according to network-based concepts, like NATO's Network Enabled Capability, henceforth NNEC [1]. The primary objective when conducting operations according to this concept, is to support the creation of a high degree of shared situational awareness among decision makers in order to obtain increased mission effectiveness. A prerequisite for achieving this, is extensive information sharing and a robust scheme for information integration, enabling decision makers to retrieve and utilize all relevant information when needed.

So far, the emphasis of the technical work on NNEC has been on how to make information available throughout the environment. However, in order for NNEC to be of use to decision makers, the challenge of establishing a robust scheme for information integration ultimately also needs to be addressed. This is the focus of the research reported on in this paper.

We present an information integration approach that combines query rewriting with data federation, and we study it in relation to an example from military evacuation planning based on live reporting of incidents over IP radio networks.

The main contribution of this paper consists in defining a novel federation strategy specifically designed for domains that share the general characteristics of this case. The most important characteristics are firstly that bandwidth is limited so the total communication costs induced by the number of HTTP request is a non-negligible factor, and secondly that the network topology is dynamic, i.e. sources may come and go. Our aim is thus to define a federation strategy that is sound and complete with respect to query answering, issues a minimal number of HTTP requests, and is compatible with run-time detection of sources.

The paper is organized as follow: In Section II we identify a list of tentative desiderata that our federation strategy should satisfy. The desiderata points to using an ontology-based data access paradigm, which is explained in Section III. Section IV outlines our solution, which is based on querying against an excerpt, or *cropping*, of the remote sources relative to an incoming query. The case study is presented in Section VI, and the main experiences drawn from the case study is presented in Section VII.

The paper assumes familiarity with W3C's Semantic Web technology stack, in particular RDF, OWL, and SPARQL. Readers not familiar with these technologies are referred to [2], [3], and [4] for an introduction.

II. Characteristics of the domain

The NNEC concept presupposes a network-based environment in which information about own and enemy units is typically distributed across several autonomous data sources contributed by coalition members. In order to support evacuation planning, these information fragments need to be integrated in order for the decision makers to obtain the highest possible degree of situational awareness. This involves tackling some idiosyncratic challenges:

- The information systems are in general semantically heterogeneous, especially in coalition operations, and cannot be accessed in a coherent and unified way,
- the underlying communication network often relies on IP radios, and is hampered by limited bandwith, latency, and limited range,

- the network topology is highly dynamic, meaning that information systems can appear and disappear at any time, and
- the shared information is mission-critical, which makes it crucial that the integration scheme yields correct and exhaustive data.

These characteristics means that we want to define an information integration approach that:

A allows a user to access available sources in a unified way,
B utilizes the available bandwith efficiently, particularly by restricting the number of HTTP requests to the remote sources,
C allows the relevant sources to be discovered at run-time, and
D guarantees the soundness and completeness of query answering.

III. Ontology-based data access

Based on the desiderata from section II, we decided to use an ontology to mitigate heterogeneities and to provide uniform access to the data. That is, we based our approach on the paradigm usually called *ontology-based data access* in which a conceptual model—the ontology—is used to express the relationship between the content of the respective sources, and to act as a single query interface towards them.

According to the W3C Web Ontology Working Group[1], an ontology defines a set of concepts or term used to describe and represent some domain of information in an abstract way that gives a formal semantics to the data in question. More specifically, an ontology gives the semantics of the data in the form of a set of logical axioms that explicate the relationship between classes of data items, and it enables computers to reason over the data as part of the process of answering a query. One particular form that this process can take, is that in which a query formulated in terms of the concepts of the ontology is successively refined until the query can be executed directly against the data. This is usually referred to as *query rewriting* and forms the basis for our approach, as explained in the next section.

Ontology-based data access is useful in all scenarios in which accessing data in a unified and coherent way is difficult. This may happen for several reasons. The data sources may have been developed for different purposes by different agencies or institutions, may not have a coherent design, and may not record similar types of information in the same manner. A well-designed ontology gives a unified view of the domain in terms of the *concepts* that are of interest to the user.

IV. Outline of approach

Our federation engine is designed to be suitable for a dynamic network topology, in accordance with our listed desiderata. To that end, sources are selected at query time based on the outcome of the reasoning process, by inspecting DNS records that are multicasted in the network (cf. section VI.). The entire federation process can thus be seen as comprised of two distinct steps. First, the query is rewritten into a query expressed directly in terms of the data according to the domain model expressed by the ontology. Next, the rewritten query is decomposed into sub-queries that yield a set of mutually exhaustive partial answers extracted from each of the selected sources.

It is not automatically the case, however, that the above mentioned steps are separable. That is, depending on the expressive power of the ontology language, reasoning may require data-access. This is not the case for the class of ontology languages that are *first-order rewritable* (cf. [5], [6]). This notion was first introduced by Calvanese et al. ([5]) in the context of the class of ontology languages called description logics. A description logic L—more generally an ontology language—is first-order rewritable if, for every ontology Σ expressed in L and a query Q , Q can be compiled into a first-order query Q_Σ that A) compiles *away* all concepts from the ontology Σ, and B) is such that given a data repository R, Q_Σ evaluated over R yields exactly the same result as Q evaluated against R and Σ.

First-order rewritable ontology languages is a crucial presupposition behind our approach. It is important precisely because it ensures that the reasoning process can be decoupled from data access. This has two hugely beneficial consequences: First, the complexity of reasoning remains unaffected as data size increases. In other words the computation time allocated to reasoning will not vary with changes in the network topology and/or availability of sources—recall that we do not assume these to stay fixed. Secondly, since reasoning can be decoupled from data access, it does not affect the federation process *per se*. That is, source selection can be performed independently of the reasoning process, which means, among other things, that a query which is run repeatedly will only have to be rewritten once.

As always there is a price to pay, though. The property of first-order rewritability imposes a serious constraint on the expressivity of an ontology language, and, as explained in section VII, must be very carefully selected in order to be able capture the salient aspects of our case. In particular, it turns out that none of the standard fragments of the W3C-endorsed ontology language OWL will do.

As regards HTTP-minimality (by which we mean keeping the number of required HTTP requests as low as possible), we found existing approaches to federated query processing not to be well suited. One of the main reasons is that they all rely on forms of join-order heuristics that tends to multiply the number of HTTP requests as a factor of the depth and number of joins: a standard distributed join algorithm will evaluate a query iteratively one triple at a time, while propagating values in a nested loop join fashion. This multiplies HTTP request in proportion to the result sets returned by evaluating each join

[1] http://www.w3.org/2001/sw/WebOnt/

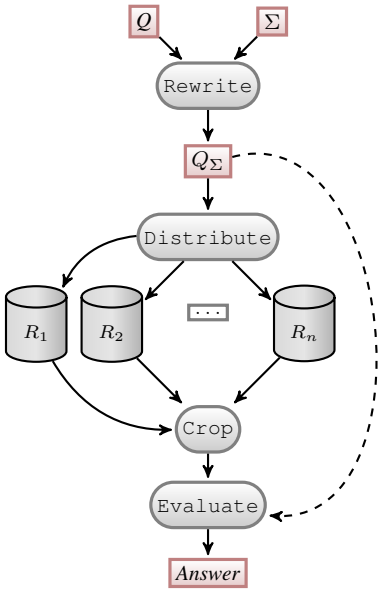


Figure 1. System overview

argument.[2] Admittedly, there are several improved versions of this algorithm on offer. The bound join technique implemented in FedX [9], for instance, groups several instances of a join argument in a single subquery using the SPARQL `UNION` construct. This reduces the number of request with a factor equivalent to the the number of instances in the grouped query (ibid.).[3] Yet, experimental evaluation shows that the number of HTTP request can still grow quite fast in the number of joins.[4][5]. What is common to all these approaches is that the number of HTTP request varies in the number of results returned by the sub-queries. It is a design goal of our approach, in contrast, to make a factor of the size of the query only.

To that end, we designed our federation engine to evaluate the query, not against the sources directly, but against an excerpt, or *cropping* as we call it, that is pulled from the sources by sending a single HTTP-request to each. Unlike traditional warehousing strategies, however, our local copy is not persisted, but exists only in-memory for the duration of the query execution process. It is essentially a snapshot of that part of the remote sources which is relevant for answering the query in question. In realistic cases, the cropping is much smaller than the total amount of data that it is extracted from (see Section V-A).

An overview of the resulting system, is shown in Figure 1: the system takes as input a SPARQL query Q, and a collection of aligned ontologies Σ, which are used by the rewriter to produce the query Q_Σ. This rewritten query is next handed to the federator component which performs service discovery at run-time (cf. Section VI) to identify live and relevant sources. Relevance here means signature overlap, where a signature is understood as a set of RDF properties. The extent of the overlap between the signature of the query and the signature of a given endpoint determines a SPARQL `CONSTRUCT` query which will be routed to that endpoint.

The `CONSTRUCT` queries are designed to adhere to a logical form which is sufficiently structured to enable us to guarantee the soundness and completeness of the query answering process wrt. the set of sources $\mathscr{R}$, as explained in more detail in the next sections. Taken together with the obvious minimality of our approach wrt. the number of HTTP requests—only a single request is sent to each source—as well as the relevance-based per-query discovery of sources, we conclude that our approach meets all our tentative desiderata A) to D).

V. Properties of the Cropping

In this section we formally define the notion of the cropping of a distributed set of sources $\mathscr{R}$ relative to a query Q, and we state its essential properties. We shall assume familiarity with SPARQL syntax and semantics (cf. [12]).

Notation. We use R_i, where i is in some index set I, to denote RDF graphs—variably referred to as sources, repositories or endpoints. A SPARQL `SELECT` query is a pair $\langle P, \vec{x}\rangle$, where P is a SPARQL graph pattern and $\vec{x}$ a vector of elements of variables. Similarly, a `CONSTRUCT` query is a tuple $\langle T, P\rangle$, where T is a basic graph pattern and P is a union of such. T will be identified with the `CONSTRUCT` block of the query, aka. the *template*, and P with the `WHERE` block, aka. the *query pattern*. We shall allow ourselves the convenience of blurring the distinction between SPARQL queries on the one hand and sets and families of triple patterns on the other. Where $Q := \langle P, \vec{x}\rangle$ is a `SELECT` query and G an RDF graph we denote the result of evaluating Q against G as $Q(G)$, and similarly for `CONSTRUCT` queries. The proofs of the claims that follow can be found in technical report [13].

As mentioned in the previous section, our approach to federation is signature-based in the sense that the RDF properties that are found in a query are used for routing different sub-queries to different endpoints. This is a common strategy (cf. [8], [9]) for which we claim no originality. Now, given a query pattern P the *relevant subset* of P in relation to a source R_i is defined as the maximal subset of P whose signature is contained in the signature of R_i. We shall denote this set as $\rho(P, i)$.

Recapitulating briefly, our federation engine is designed to be HTTP-minimal, as well as sound and complete wrt. to query answering over the selected sources. A strategy that supports all three is to execute the query against an *in-memory representation* of the remote sources rather than against the sources themselves. More specifically our federation engine routes a single `CONSTRUCT` query to each of the selected sources—achieving HTTP minimality—whereas the logical form of this construct query is defined in such a manner as

[2] DARQ [7] and SPLENDID [8] both implement a version of this algorithm.
[3] Similar techniques are considered in [8] and [10]
[4] See e.g. the results for FedX on Life Sciences 3 query from the FedBench suite.
[5] Another notable optimization is the star-shaped pattern technique of [11]. Numbers of requests are however not reported in this study.

to guarantee that the answer to the query assembled from the selected sources is both correct and complete with respect to those sources. Here soundness and completeness means that if $\mathscr{R}$ is a set of sources selected for federation, then the answer that the federator provides to a query Q should be exactly the same as the one that would be obtained were Q to be evaluated conventionally over a single repository holding the union of the data sets in $\mathscr{R}$. To the best of our knowledge, our strategy is currently the only one that guarantees that this is the case.

The logical form in question is in turn defined by distinguishing between *exclusive* and *non-exclusive* triples in a query pattern P. Exclusive triples are those that are satisfied, if at all, at one endpoint only. Non-exclusive triples, on the other hand, may be satisfied by two or more. Exclusive triples can safely be grouped together and executed against the source for which it is exclusive in as a single conjunctive pattern. Non-exclusive triples, however, must be shipped to the remote sources as separate `UNION` clauses. This holds even if a group of triple patterns are relevant to exactly the same sources since an answer to the original query may require joining triples across these sources. This gives rise to the following definition of *the set of clauses induced by P and R_i*:

Definition 1 (Clause set): For R_i a source and P a query pattern: $s(P,i) := \{\epsilon(P,i)\} \cup \{\{t\} : t \in \rho(P,i) \setminus \epsilon(P,i)\}$

Here $\epsilon(P,i)$ denotes the exclusive group of a pattern P relative to R_i.

Now, the basic idea behind our federation strategy is to use the set of clauses induced by P and R_i to define a `CONSTRUCT` query that extrapolates the part of R_i that is relevant for answering P. The most straightforward way to do that may seem to be to use the clause set itself as a query pattern, whilst using the set-theoretic *union* of its elements as a template. Call this the *naive* strategy. Interestingly, the naive strategy, whilst complete, is *not* sound. Consider the following rather abstract example:

Example 1: Let G be the RDF graph containing only the two triples $\mathtt{s} := (\mathtt{c1}, \mathtt{p}, \mathtt{d1})$ and $\mathtt{t} := (\mathtt{c2}, \mathtt{q}, \mathtt{d2})$, and assume a clause-set $\{\{(?\mathtt{s}, \mathtt{p}, ?\mathtt{o})\}, \{(?\mathtt{s}, \mathtt{q}, ?\mathtt{s})\}\}$. The corresponding naive `CONSTRUCT` query is:

```
CONSTRUCT {?s q ?o. ?s p ?o.}
WHERE {{?s p ?o} UNION {?s q ?o}}
```

Executing this query against G will produce a graph containing the triple $(\mathtt{c1}, \mathtt{q}, \mathtt{d1})$.

The example shows that the naive strategy may create bindings in the resulting graph that do not exist in the graph that is queried. To counteract this effect it is necessary to standardize apart the elements of the clause-sets before using taking the union and using it as a `CONSTRUCT` template. To this end we introduce the notion of a *separation function*:

Definition 2 (Separation function): Let $S := \{c_1, \ldots, c_n\}$ be a clause set, and let σ_i be a uniform substitution of variables for variables in c_i. A separation function f for S is a function s. t. 1) $f(S) = \{\sigma_1(c_1), \ldots, \sigma_n(c_n)\}$, and 2) $\sigma_j(?x) \neq \sigma_k(?x)$ for every $?x \in dom(\sigma_j) \cap dom(\sigma_k)$.

Our `CONSTRUCT` queries now become:

Definition 3: For a set of sources $\mathscr{R} := \{R_i\}_{i \in I}$ and a query pattern P: $\mathscr{C}(P,i) = \langle \bigcup f(s(P,i)), f(s(P,i)) \rangle$ where f is some separation function for $s(P,i)$.

Example 2: Suppose we have two endpoints `JOCWatch` and `MedWatch`,[6] and the following conjunctive graph pattern P:

```
?mission medics:missionType medics:Rescue.
?mission medics:jocWatchIncident ?incident.
?incident jocw:status ?stat.
```

Suppose further that each property prefixed by `medics` belongs to the signature of `MedWatch`, that each property prefixed by `jocw` belongs to the signature of `JOCWatch`, and that the `jocw:status` property belongs to both. The queries that are routed to the respective endpoints are then:

MedWatch:

```
CONSTRUCT {
 ?_1 medics:missionType medics:Evac.
 ?_1 medics:jocwIncident ?_2.
 ?_3 jocw:status ?_4. } WHERE {
 { ?_1 medics:missionType medics:Evac.
   ?_1 medics:jocwIncident ?_2.}
   UNION
 { ?_3 jocw:status ?_4.}}
```

JOCWatch:

```
CONSTRUCT {
 ?_1 jocw:instigator ?_2.
 ?_3 jocw:status ?_4.} WHERE {
 { ?_1 jocw:instigator ?_2.}
   UNION
 { ?_3 jocw:status ?_4.}}
```

The cropping may now be defined as follows:

Definition 4 (Cropping): Put $Q := \langle P, \vec{x} \rangle$ and $\mathscr{R} = \{R_i\}_{i \in I}$. Then $\mathscr{A}^{\mathscr{R}}_Q := \bigcup_{i \in I} \mathscr{C}(P,i)(R_i)$.

We now have:

Theorem 1 (Soundness/Completeness): Let $\mathscr{R}$ be any set of sources, then $Q(\bigcup \mathscr{R}) = Q(\mathscr{A}^{\mathscr{R}}_Q)$ for any `SELECT` query Q.

Note that here Q is the `SELECT` query that is being posed to the system, whereas $\mathscr{A}^{\mathscr{R}}_Q$, i.e. the cropping, is the result of assembling the results of the `CONSTRUCT` queries that are required for providing an excerpt guaranteed to answer it.

As regards time complexity, since every `CONSTRUCT` query $\mathscr{C}(P,i)$ is in union normal form, Corollary 1 of [12] immediately entails that the cropping can be built efficiently. In our actual implementation, the `CONSTRUCT` queries that are allocated to the respective endpoints are, moreover, all executed in parallel, so time is not a precarious measure.

A. Restricting the size of the cropping

Although time is not a precarious measure, the size of result sets quickly becomes an issue. The `CONSTRUCT` queries that are passed around to the remote endpoints, if not constrained,

[6] These systems are described in VI

may well distribute a triple pattern $(?\texttt{a}, \texttt{rdf:type}, ?\texttt{b})$ to all remote endpoints, in effect requesting huge chunks of the data contained in each.

Now, there is no need in our approach for join-ordering heuristics in the conventional sense, since, per the approach, joins are either executed remotely, or executed locally by a standard query processing engine after the cropping has been built. Rather, what we do, is to build the cropping incrementally by assessing the relative selectivity of triple patterns and processing the most selective ones first. We can only describe this procedure in general outline here:

The selectivity of a triple pattern may be assessed along several dimensions. For instance, studies show that a triple pattern with a literal in object position will usually be more selective than one with a URL in the same position [14]. Moreover, triple patterns can be ordered in a plausible sequence of decreasing selectivity based on the distribution, and position, of variables in the pattern (? denotes a variable):

$(s,p,o) \prec (s,?,o) \prec (?,p,o) \prec (s,p,?) \prec (?,?,o) \prec (s,?,?) \prec (?,p,?) \prec (?,?,?)$

The entire set of heuristic rules that we have used in our solution can be found in [14].

Now, the idea is to build the cropping in layers by employing the following three-step procedure: 1) construct the graph corresponding to the most selective patterns pertaining to each endpoint 2) extract variable bindings from the cropping so far, and 3) pass them on to the next iteration as constraints for the next round of queries.

Step 2, the extraction of variable bindings, is realized by re-using the triple patterns as `SELECT` queries that are evaluated against the cropping as it exists so far, whereas the propagation of values from one layer to the next is realized with the `VALUES` feature of SPARQL 1.1, which allows a set of bindings to be shipped with a query in order to constrain the answers.

Our procedure is designed to treat each exclusive group as an atomic unit, since exclusive groups are likely to be more selective as a *set*. For the same reason, they are given maximum priority, That is, the first layer of the incremental construction of the cropping consists of the result of executing the exclusive groups as `CONSTRUCT` queries. The subsequent layers are then constructed from the non-exclusive patterns by rating them according to the heuristic criteria.

This procedure is sufficient to ensure that very unconstrained patterns such as $(?\texttt{s}, ?\texttt{p}, ?\texttt{o})$ will be processed late, when bindings are available for some of its variables. It can also be tuned to give low priority to predicates from existing RDF vocabularies that are known to have a low selectivity rate, such as e.g. $\texttt{rdf:type}$ or $\texttt{dcterms:title}$. The procedure preserves the soundness and completeness of the cropping wrt. the underlying sources, and, although the number of HTTP request is no longer minimal it is constant in a small factor of the number of triple patterns in the original query.

VI. Case

The case we use to exemplify our approach is based upon the following scenario: A military analyst is monitoring planned medical evacuation flight missions, and is on the lookout for missions that might be threatened by enemy activity. If this is the case, she is also interested in finding friendly units able to counter the particular threat. More specifically, the information requirement is the following: *Find all medical evacuation missions and friendly units such that a) the mission can be classified as being threatened; and b) that the friendly unit can handle the specific type of threat that the enemy poses.*

Normally, in order to obtain an answer to this information requirement, the analyst has to keep an eye on several systems, as information about evacuation flights and information about enemy activity are usually not kept in the same system. With the aid of a system as outlined so far in this paper, however, the analyst can pose a query formulating what she is looking for and let the information integration system take care of the rest.

To evaluate the approach and the case outlined above, we conducted an experiment at NATO CWIX 2013[7] set as close as possible to the dynamic and multinational environment of NNEC. The experiment involved three operational information systems: 1) `JOCWatch`, information on incidents of relevance to the command in an event log, 2) `MedWatch`, a system for medical mission tracking designed to support the planning, logging and monitoring of medical evacuation missions, and 3) `Track Source`, a unit tracking service providing times-tamped geopositional information regarding friendly units in the field. The information in `MedWatch` and `JOCWatch` were made available through SPARQL endpoints by using D2R [15], while the Track Source service had a native SPARQL interface. In addition, all sources were supplied with a service description according to the SPARQL 1.1 specification, and each source made available its ontology at an URL described in the service description.

Our prototype system performed service discovery using

- mDNS[8] for broadcasting and discovering the presence of information sources,
- DNS-SD[9] for high-level description of the source in terms of pointers to query endpoint location and content description location, and
- the SPARQL 1.1 service description and VoID[10] vocabularies for describing source content.

This approach addressed the NNEC needs as outlined in section II, and had the advantage that it was independent of a central registry, thus eliminating the issue of network

[7] Coalition Warrior Interoperability eXploration, eXperimentation and eXamination, eXercise – an annual NATO event aimed at improving alliance-wide interoperability

[8] http://www.multicastdns.org/

[9] http://www.dns-sd.org/

[10] http://www.w3.org/TR/void/

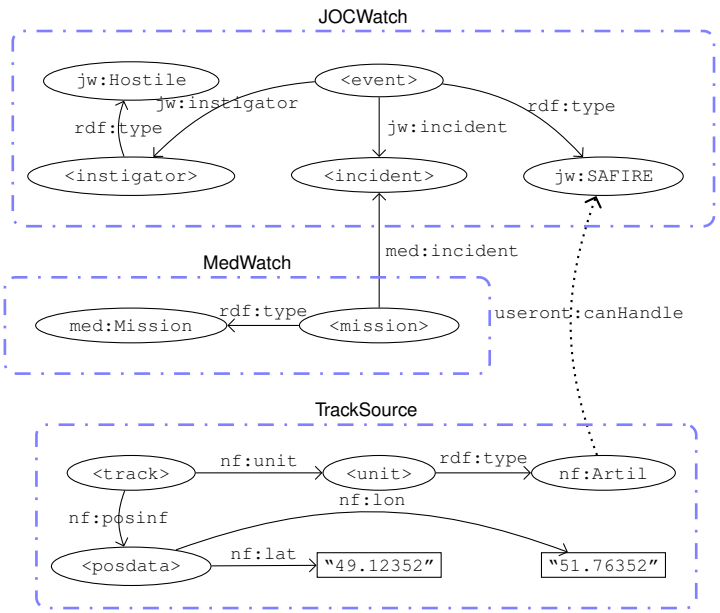


Figure 2. Conceptual relationship between data sources

Concept	Definition
ThreatenedMission	MedWatch missions that are related to a ThreateningIncident
ThreateningIncident	All JOCWatch incidents that are related to a ThreateningEvent
ThreateningEvent	All JOCWatch events that are both a MilitaryOperation (from the JOCWatch ontology) and a HostileEvent
HostileEvent	All events that has a HostileInstigator
HostileInstigator	All event participants that are classified as being hostile.
Relation	**Definition**
canHandle	A relation between a military unit type and the type of events those units types are equipped to handle.
hasEvent	If a mission involves an incident, and there exists an event that belongs to the same incident (inverse property), then the event is also related to the mission

Table I. RELEVANT DEFINITIONS IN THE USER ONTOLOGY

```
SELECT ?mission ?unit
WHERE{
   ?mission a useront:ThreatenedMission.
   ?mission useront:hasEvent ?event.
   ?event a useront:ThreateningEvent.
   ?event wgs84:lat ?elat.
   ?event wgs84:long ?elong.
   ?unit useront:canHandle ?event.
   ?unit useront:hasPosition ?pos.
   ?pos wgs84:lat ?ulat.
   ?pos wgs84:long ?ulong.
}
```

Figure 3. SPARQL query representing the information request

fragmentation.

The relationship between the information sources in our experiment is illustrated in Figure 2. In the figure we see that MedWatch missions are (potentially) related to JOCWatch events through a shared incident. Furthermore, the JOCWatch events are typed according to category e.g. as a SAFIRE event, which is an event that involves hostile surface-to-air fire. In the figure we also see that units in the Track Source are typed (e.g. as Artillery), and that it contains positional data. Additionally, in the figure, a stipulated line is drawn from Artillery to SAFIRE, indicating that, units of the former kind are equipped to counter those of the latter. This relation does not actually belong to the data, but is defined in the user ontology useront, and is key to the formulation of the user's information need.

The experiment included four main ontologies: a JOCWatch ontology, a MedWatch ontology, a Track Source ontology, and an ontology containing the concepts used by the user of the system.

In this particular case, the user ontology is derived from and expresses the data models of the respective sources. We are thus assuming that, although available sources may come and go, we know their data models. This is not overly unrealistic, since NATO-wide standardization is part of the NNEC concept. The assumption means that we do not have to match ontologies at run-time. A user ontology less tightly coupled with the source ontologies and run-time ontology matching is something we plan to look more into in future work.

Given the user ontology, the information requirement described earlier can now be expressed by the query in Figure 3. [11]

Here ThreateningEvent, hasEvent, canHandle, and hasPosition are terms specific to the user's vocabulary, see table VI. Posing this query to any of the information sources would not return any answers. In our experiment, this query was decomposed and distributed as per the approach outlined earlier.

The main motivation behind this experiment was to test whether it is feasible to provide a decision maker with the means to request information using her own terms and without presupposing detailed knowledge about a fixed set of sources. This creates a coupling between the requesting system and the information sources that is loose enough to adapt to a changing network topology, something that should be highly relevant in the NNEC environment. Our strategy of combining once-per-query federation with rewriting worked well for our sample case, and proves that the idea is sound in general outline. To be sure, if the system is to to scale well—both in terms of efficiency and usability—there are some serious issues that need to be addressed having to do with the expressiveness of the ontology language and the complexity of reasoning in it. We record some findings in the next section.

[11] In reality, we apply filtering of friendly units based on distance from events using the haversine formula and a threshold value. As this does not contribute to understanding the general approach we have left it out of the example.

VII. Experiences and Observations

As explained in section IV, it is an essential presupposition of our approach to federation that the ontology that is used to provide access to the underlying sources be expressed in a first-order rewritable language. This is necessary in order to separate reasoning from source selection, thus making the system able to adapt to a dynamic network topology by selecting sources at run-time.

Yet, it is not given that the structure and relationship between the sources that we selected for our case-study, as illustrated in Figure 2, can in fact be expressed in a first-order rewritable language.

Choosing an ontology language in the DL-Lite family of description logics would have been natural for several reasons: First,these languages are specifically designed to stay within the boundaries of first-order rewritability. Secondly, DL-Lite forms the basis of the W3C-endorsed QL language profile of OWL 2, and so has an XML serialization, and enjoys the status of an official recommendation. Finally, several efficient rewriters already exist for the DL-Lite family of languages, which, if we could use them, would of course leverage the burden of implementing our own federation engine.

As it turns out, however, our case cannot be expressed in any of the standard OWL2 profiles, nor in any other description logic we are currently aware of. This is due to a combination of features exemplified by the structure of our sources. Referring to Figure 2 there are mainly two sources of expressive complexity:

1) As indicated by the stipulated arrow labelled `useront:canHandle` connecting `TrackSource` to the `JOCWatch` database, we wish to add axioms to the ontology that classify which kind of vehicle or unit that is equipped to counter which kind of hostile event—for instance artillery in the case of a surface-to-air attack. Encoding this knowledge in the ontology is necessary in order to enable the user to query the sources for available military support within a given diameter from a threatened position. However, it requires that we be able to state in the ontology that certain combinations of unit types and events constitute sufficient conditions for the unit and event in question to stand in the `canHandle` relation. Stated more formally, we need to have axioms of the form $\forall x \forall y. Artillery(x) \wedge SAFIRE(y) \rightarrow canHandle(x, y)$, for all the appropriate combinations of units and events.

2) Presupposing that the `canHandle` relation has been axiomatized, we further need to express in the ontology that finding the position of a unit involves traversing the `TrackSource` graph from the reported latitudes and longitudes through the relations `nf:posinf`, `nf:unit` and `rdf:type` via `useront:canHandle` to an associated hostile event. This is a fairly long and intricate path that requires traversing relations forwards as well as backwards (i.e. traversing the inverse of the relation).

The problem with 1) is that it has a binary predicate in the conclusion. For that reason, it cannot be expressed as a class inclusion axiom. A description logic axiom is either a class axiom or a relationship axiom (aka. role axiom) but cannot be a mix of the two. Indeed, a class inclusion axiom—irrespective of the particular brand of description logic that is being used—cannot, by design, express cross-references between antecedent and consequent in two or more variables, as our axioms require.

Description logics typically allow us to state axioms like the following (in description logic notation):

- $\texttt{Artillery} \sqsubseteq \exists \texttt{canHandle.SAFFIRE}$
- $\texttt{Artillery} \sqsubseteq \forall \texttt{canHandle.SAFFIRE}$

At first glance, these may seem to come close to what we wish to say, but that is not really the case. The first says that an artillery unit can handle *some* surface-to-air-fire event, but it does not identify the event. The second says that an artillery unit can *only* handle surface-to-air-fire events, although it may not be able to handle *all* of them. What we wish to say though us that *all* artillery units can handle *all* surface-to-air-fire events.

Taking stock, there is thus, to the best of our knowledge, no first-order rewritable description logic capable of expressing the structure and interrelationship between our selected sample of military information sources. As it turns out, though, there is a different family of ontology languages altogether that *is* sufficiently expressive for our needs, namely the family of *general existential rules* aka. *existential datalog* [6], more specifically the language of *weakly recursive datalog*. Weakly recursive datalog is strictly more expressive than any first-order rewritable description logic—and more importantly, it is sufficiently expressive to express 1) and 2) above, thus capturing the salient features of our case. Our federation engine is therefore equipped with a rewriter that expects an ontology to be encoded in weakly recursive datalog.

Although, this choice is more or less forced upon us by the characteristics of the case, it does not mean of course that the choice of recursive datalog as our ontology language does not come with its own set of drawbacks. First of all, existential datalog in general does not currently have the kind of institutionalized support that the OWL family of languages enjoys. Secondly, and much for the same reason, it has far less endorsement from the software industry in terms of tool support.

In fact, we could not find an existing rewriter for weakly recursive datalog, and therefore had to build one from scratch. Alas, implementing a correct rewriter does not entail that one has implemented an efficient one, and although queries over weakly recursive datalog ontologies are first-order rewritable, the size of the rewriting itself may be exponential in the size of the original query. Thus, without a considerable amount of research being devoted to optimization, the rewriter is not

likely to perform well for any large class of cases. Theoretical results are encouraging, though. In particular, results from [16] shows that there is a minimal rewriting of any query over a set of weakly recursive datalog rules. Computing such a minimum, however, remains a topic for future research.

VIII. Related Work

Several studies have addressed the problem of decomposing a SPARQL query into sub-queries that can be allocated to a distributed set of remote sources. Notable examples include [17], [7] [18], [19], [20], [9], [11] and [8]. All of these studies belong to what we would call the join-order heuristics paradigm, and, unlike the present paper, none gives particular attention to establishing framework that is both sound/complete and request-minimal. Moreover, the listed reports focus exclusively on federating queries that are expressed directly in terms of the data. To the best of our knowledge there are very few contributions that address the question of how to combine query federation with reasoning, where reasoning cuts across several sources.

IX. Conclusion

In this paper we have established a sound, complete and request-minimal baseline for query federation. Our approach is signature-based and compatible with a run-time selection of sources. It is therefore particularly suitable for domains that are characterized by low bandwidth and a dynamic network topology. We have further described an example from military evacuation planning to illustrate the usefulness of the approach. In order to mitigate the heterogeneities between sources, as well as to present the data in a vocabulary that is familiar to the user, we found it expedient to use an ontology to provide a unifying layer above the information sources. Any incoming query is therefore rewritten according to the ontology before being passed on to the sources. However, we have that ontology-based data access is not coherent with our federation strategy unless the ontology is formulated in a language that is first-order rewritable such that reasoning can be decoupled from data access. In realistic cases like ours, one quickly transcends the expressive capabilities of familiar first-order rewritable OWL fragments such as OWL2-QL. In our case we overcame this limitation by resorting to a decidable fragment of existential datalog.

Acknowledgments

The NATO systems participating in the experiments reported in this paper was made available to us by the NATO C3 Agency.

References

[1] P. Bartolomasi, T. Buckman, A. Campell, J. Grainger, J. Mahaffey, R. Marchand, O. Kruidhof, C. Shawcross, and K. Veum, "NATO Network Enabled Capability Feasibility Study, Version 2.0," NATO C3 Agency, Tech. Rep., October 2005.

[2] W3C, "RDF Primer," http://www.w3.org/TR/rdf-primer/, February 2004.

[3] ——, "OWL 2 Web Ontology Language Primer," http://www.w3.org/TR/2009/REC-owl2-primer-20091027/, October 2009.

[4] ——, "SPARQL 1.1 Query Language," http://www.w3.org/TR/sparql11-query/, March 2013.

[5] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodriguez-Muro, and R. Rosati, "Ontologies and Databases: The DL-Lite Approach," in *Semantic Technologies for Informations Systems*. Springer, 2009.

[6] G. Gottlob, G. Orsi, and A. Pieris, "Ontological Queries: Rewriting and Optimization (Extended Version)," *Computing Research Repository*, vol. abs/1112.0343, 2011.

[7] B. Quilitz and U. Leser, "Querying distributed RDF data sources with SPARQL," in *Proceedings of the 5th European Semantic Web Conference (ESWC '08)*, 2008.

[8] O. Görlitz and S. Staab, "SPLENDID: SPARQL Endpoint Federation Exploiting VOID Descriptions," in *Proceedings of the 2nd International Workshop on Consuming Linked Data (COLD 2011)*, 2011.

[9] A. Schwarte, P. Haase, K. Hose, R. Schenkel, and M. Schmidt, "FedX: Optimization Techniques for Federated Query Processing on Linked Data," in *Proceedings of the 10th International Semantic Web Conference (ISWC 2011)*, 2011.

[10] J. Zemanek and S. Schenk, "Optimizing SPARQL Queries over Disparate RDF Data Sources through Distributed Semi-Joins," in *International Semantic Web Conference (Posters and Demos)*, ser. CEUR Workshop Proceedings, vol. 401, 2008.

[11] G. Montoya, M.-E. Vidal, and M. Acosta, "A heuristic-based approach for planning federated sparql queries." in *COLD*, ser. CEUR Workshop Proceedings, J. Sequeda, A. Harth, and O. Hartig, Eds., vol. 905. CEUR-WS.org, 2012.

[12] M. Arenas, C. Gutierrez, and J. Pérez, "Foundations of RDF Databases," in *Reasoning Web. Semantic Technologies for Information Systems*. Springer, 2009.

[13] B. J. Hansen, J. Halvorsen, and A. Stolpe, "Information integration experiment at NATO CWIX 2012," Norwegian Defence Research Establishment (FFI), FFI/RAPPORT-2012/01543, 2012.

[14] P. Tsialiamanis, L. Sidirourgos, I. Fundulaki, V. Christophides, and P. Boncz, "Heuristics-based query optimisation for sparql," ser. Proceedings of EDBT '12. ACM, 2012.

[15] C. Bizer and R. Cyganiak, "Publishing Relational Databases on the Semantic Web," http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/, August 2009.

[16] C. Civili and R. Rosati, "A broad class of first-order rewritable tuple-generating dependencies," in *Datalog in Academia and Industry*, ser. Lecture Notes in Computer Science, P. Barceló and R. Pichler, Eds. Springer Berlin Heidelberg, 2012, vol. 7494, pp. 68–80.

[17] M. Acosta, M.-E. Vidal, T. Lampo, J. Castillo, and E. Ruckhaus, "Anapsid: An adaptive query processing engine for sparql endpoints." in *Proceedings of the 10th International Semantic Web Conference (ISWC 2011)*, 2011.

[18] C. Basca and A. Bernstein, "Avalanche: Putting the spirit of the web back into semantic web querying," in *ISWC Posters&Demos*, 2010.

[19] A. Harth, K. Hose, M. Karnstedt, A. Polleres, K.-U. Sattler, and J. Umbrich, "Data summaries for on-demand queries over linked data," in *Proceedings of the 19th international conference on World wide web*, ser. Proceedings WWW '10. New York, NY, USA: ACM, 2010, pp. 411–420.

[20] Y. Li and J. Heflin, "Using reformulation trees to optimize queries over distributed heterogeneous sources," in *Proceedings of 9th the International Semantic Web Conference (ISWC)*, ser. Proceedings ISWC'10. Springer-Verlag, 2010.

Navigation Assistance Framework for Emergencies

Paul Ngo
Department of Computer Science
George Mason University
4400 University Drive, MS 4A4
Fairfax, Virginia 22030
Email: pngo1@gmu.edu

Duminda Wijesekera
Department of Computer Science
George Mason University
4400 University Drive, MS 4A4
Fairfax, Virginia 22030
Email: dwijesek@gmu.edu

***Abstract*—Emergencies occur every day at unexpected times and impact our lives in unimaginable ways. In any emergency situation, there are two type of victims: direct victims and indirect victims. Both will have their current plans disrupted in order to deal with the emergency. Federal, State, and Local governments have established a 911 system to assist direct victims. However, there is still lack of assistance provided to the indirect victims. In this paper, we propose a Navigation Assistance Framework that allows emergency organizations to provide emergency information that can assist victims navigating out of the emergency area and reaching their intended destinations in a reasonable amount of time. We develop an emergency prototype ERSimMon to simulate this capability in a small scale to show the effectiveness of the proposed solution. In addition, we develop the Emergency Response Application (ERApp) for a smart phone platform, which intercepts the enhanced Commercial Mobile Alert System (CMAS) broadcast message, displays the user's location with respect to the emergency location on the map and provides navigational assistance and recommend actions to help the user navigate out of ongoing emergencies.**

I. Introduction

According to the Out-of-State and Long Commutes Survey 2011 [12], 8.1 percent of U.S. workers had commutes of 60 minutes or longer. In addition, 61.1 percent of the workers drove to work alone. Americans spend significant amounts of time, on the average of 25 minutes [13] in their vehicles on the road to go from home to work on a normal working day.

Added to average commute time, local emergencies such as car accidents, road construction, inclement weather, etc. may add extra delays into the average commute time. Commuters have to adjust to these unexpected delays on a case-by-case basis. Consequently, they may have to shift their schedule or rearrange appointments and meetings to accommodate for the time lost sitting in traffic. Sometimes, cancellations and delays are unavoidable. According to a poll conducted by ABCNews on traffic in the United States [14], the average commute time on a bad day for Americans is 46 minutes.

Clearly, dealing with unexpected delays is a major concern for commuters. We address this concern with two approaches. The first approach is to provide commuters with navigational assistance that offers alternative routes to their destinations in order to avoid an impending emergency and its affected area. This may be a great help to commuters who are not familiar with an area or who waste time sitting in traffic. The second approach is to provide commuters with relevant emergency advice based on the type of the emergency.

In 2006, the Federal Government established a Worker Adjustment and Retraining Notification (WARN) Act that supported the research and development of Common Mobile Alert System (CMAS) [15]. The proposed CMAS system utilizes existing commercial telecommunication infrastructures to broadcast emergency alerts and warnings to a specified geographic area. We have extended the usability of CMAS to broadcast alerts to small-scale local emergencies [2]. We convey these local emergencies by sending the GPS location of the emergency and the affected area measured by the radius from the emergency GPS location to mobile users' devices. We also enhanced the original CMAS limitation on the message size of 90 readable characters [1]. Both of these CMAS enhancements allow local emergency information to be broadcast to mobile users more effectively.

To provide relevant navigational assistance to mobile users in a variety of emergencies, from the most dynamic, like a tornado or hurricane, to the least changing such as construction road blocks, we need the most up-to-date information regarding the emergency. We propose a Navigation Assistance Framework (NAF) to set a foundation for possible future works. The NAF acts as a central hub, which collects relevant emergency information and distributes it to registered instances of our smart phone application, ERApp. This development is aligned with the Dynamic Mobile Application initiative from the US Department of Transportation [20], which can be adapted to cars to alert drivers when approaching work-zones or construction sites [21].

The rest of the paper is organized as follows: section II discusses NAF requirements and some supporting use cases. Section III discusses the NAF design and implementation. Section IV discusses results of our experiments. Section V describes related works and we conclude in section VI.

II. Navigation Assistance Framework Requirements and Use Cases

In this section, we specify some requirements and objectives that organizations may implement in their processes and operations in order to provide emergency information to other trusted organizations. A requirement contains the word "shall" and is identified by the letters "**R**". An *Objective* is a feature

or function that is desirable, but not mandatory. An Objective contains the words ”it is desirable” and is identified by the letters ”**O**”.

R#1: There shall be a way to provide current information about any impending emergency.

R#2: There shall be a way to provide directions to avoid the impending emergency.

O#1: It is desirable that users provide daily events in their calendars and expose them to the trusted entity in order to provide relevant and immediate actions during time of crisis.

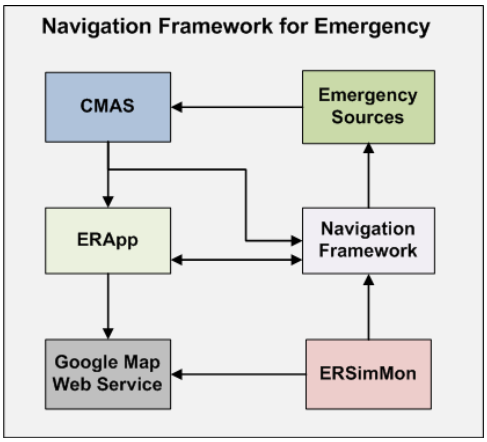


Fig. 1. Navigation Assistance Framework High-Level Components

A. Use Cases

In this subsection, we describe use cases that are derived from the above requirements.

Use Case 1: A driver with an ERApp running on his hand-held device drives to work as a part of his regular routine.

The following two use cases occur when the driver receives a CMAS message informing him that there is an emergency in the area. ERApp appears on his hand-held device, showing the location of the ongoing tornado, his location and the work location. He then determines that:

Use Case 2: his route to work has not been impacted by the ongoing tornado.

Use Case 3: his route to work is impacted by the ongoing tornado.

The first use case illustrates a sunny day scenario where drivers don’t encounter any problems on the road that prevent them from arriving at work on time. However, traffic accidents and natural emergencies such as tornados, heavy rains, blizzards, snow storms, hail, or other severe weather conditions would prevent drivers from arriving at work on time. Delays caused by these emergencies can be up to hours. The second and third use cases illustrate that an emergency has occurred in the area. In this case, we illustrate with an impending tornado because the tornado is a medium scale emergency and its movement can be tracked by the National Weather Center (NWC) [19]. The NWC then can provide the Navigation Assistance Framework crucial data such as the direction and the speed of the tornado. We can then use these data to calculate and estimate the impact further.

III. Navigation Assistance Framework Architecture and Implementation

We describe major components and the functionality of the Navigation Assistance Framework (NAF) in this section. First, we provide a high-level description of each component and its function in the overall architecture.

A. Architecture Components

Figure 1 shows the high-level architectural components for the Navigation Assistance Framework. It consists of Emergency Sources, Commercial Mobile Alert System (CMAS) [15], Navigation Assistance Framework, Emergency Response Application (ERApp), Emergency Response Simulation Monitor, and Google Map Services.

Emergency Sources are emergency systems that have the capability to monitor the progression of an emergency and to provide updates if needed by other systems. These emergency systems can expose their emergency information as a service. We provide a set of interfaces in table II that can be implemented by Emergency Sources. For example, the National Hurricane Center [18] is considered one of the Sources for emergency information. The Emergency Sources push the most up-to-date emergency data to the CMAS through a web service connection as indicated by the arrow going from the Emergency Sources to CMAS in Figure 1. The CMAS operator will generate the broadcast message based on the emergency data and broadcast it. We have proposed a few enhancements [1], [2] to improve the content of the broadcast message and the area effected by an emergency. The CMAS broadcasts 90-character text messages of emergencies to all mobile devices through ERApp, a mobile emergency application installed on the users’ mobile devices (how the ERApp is certified and installed on mobile devices is beyond the scope of this paper). In addition, the CMAS pushes the emergency data to the NAF as indicated by the arrow going from the CMAS to the NAF through a web service connection in Figure 1.

The Navigation Assistance Framework provides a set of interfaces that Emergency Sources need to implement and acts as a listener to the emergency data and advice policies. Whenever needed, the Navigation Assistance Framework pulls the most up-to-date emergency data and advice policies from the Emergency Sources as indicated by the arrow going from the Navigation Framework to Emergency Sources in Figure 1. The ERApp can make an advice policy update request to the NAF when the ERApp detects that the user is in motion during an ongoing emergency as indicated by the arrow going both directions from the ERApp to the NAF in Figure 1. The ERApp uses the Google Map Web Services to display a user’s

position with respect to the occurring emergency as indicated by the arrow going from the ERApp to Google Map Web Services in Figure 1. The ERApp applies the advice policies to see if the current status of users' behaviors satisfy the conditions on the policy and displays the emergency advice recommended by the policy. For example, the advice policy can say that if the user at rest is 3000 meters away from the center of an emergency, the user needs to consider teleworking for the day. Figure 2 shows such a sample policy written in XACML [17]. XACML policy language answers yes or no to the access control request based on some conditions stated in a policy. Consequently, XACML is not capable of providing emergency advice. Therefore, the NAF uses XACML to evaluate conditions in emergency advice policies given by Emergency Sources before advising users.

In general, these arrows presented in the figure 1 are defined by either push, pull or both push and pull web services. Implementation and the hosting of these web services are beyond the scope of this paper.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Policy xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-policy-schema-os.xsd"
  RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides"
  Version="1.0" PolicyId="EmergencyAdvicePolicy2">
  <Target>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Consider teleworking today.</AttributeValue>
          <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
                                       DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </ResourceMatch>
      </Resource>
    </Resources>
    <Actions>
      <Action>
        <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">access</AttributeValue>
          <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                                     DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </ActionMatch>
      </Action>
    </Actions>
  </Target>
  <Rule Effect="Deny" RuleId="comparing_utoedistance">
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-one-and-only">
          <SubjectAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#integer"
                                      AttributeId="utoedistance" />
        </Apply>
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">3000m</AttributeValue>
      </Apply>
    </Condition>
  </Rule>
  <Rule Effect="Deny" RuleId="matching_ertype">
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
        <SubjectAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                                    AttributeId="ertype" />
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Tornado</AttributeValue>
      </Apply>
    </Condition>
  </Rule>
  <Rule Effect="Deny" RuleId="is_moving">
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:boolean-equal">
        <SubjectAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#boolean"
                                    AttributeId="isInMotion" />
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#boolean">true</AttributeValue>
      </Apply>
    </Condition>
  </Rule>
  <Rule Effect="Permit" RuleId="rule_permit_all"/>
</Policy>
```

Fig. 2. Emergency Advice XACML Policy

The Emergency Simulation Monitor (ERSimMon) uses the Navigation Assistance Framework to simulate an emergency and the people who are trying to navigate through it. The ERSimMon uses the Google Maps API web services to query the list of emergency constraints and road congestion information in order to suggest the best routes that the user can take to reach his destinations. This pulling connection is indicated by the arrow going from the ERSimMon to the Google Map Web Services in Figure 1.

For these components to work seamlessly, we need to make a couple of enhancements to the Google Maps API web service and Emergency Source web service. These enhancements allow the Navigation Assistance Framework to be used in the most effective way and expose its full capabilities. Here is the list of enhancements:

First, we propose an enhancement to the Google Maps API web services [7] to include a list of constraints and road blocks. The original URL to get directions from Google is: http://maps.googleapis.com/maps/api/directions/xml?origin=[]&destination=[]&sensor=[true|false] where the *origin* parameter specifies the origination address. The *destination* parameter specifies the destination address. The *sensor* parameter indicates that the directions request comes from a device with a location sensor. There are a few optional parameters such as mode= [driving|walking|bicycling|transit], waypoints, alternatives= [true|false], avoid= [tolls|highways], language, units, region, departure_time, and arrival_time. None of these parameters provide the directions to avoid emergency road blocks or help navigate around pending emergencies. At best *alternative* parameters provide several routes to the destination, without any guarantee that these routes will avoid emergency road blocks or the pending emergency.

Therefore, our enhancement adds one parameter *eblocks* into the Google Maps API web service, which gives the GPS location and the radius of the blocking area. The parameter has three values: latitude, longitude, and radius. For example: eblocks=38.8462236,-77.3063733,500m. In this example, we indicate that the emergency occurs in Fairfax, VA which has the GPS location of 38.8462236,-77.3063733 and we should avoid all the roads within 500 meters of that particular location.

The NAF doesn't depend on the Google Map enhancement to provide alternative routes in order to avoid the area affected by an emergency. But in this paper, we show how Google could implement this enhancement as we describe below. The NAF can use major routes and intersections as preexisting points and build a directed path using the Shortest Path (in time and distance) algorithm [25] to determine the path to the destination. For every connecting point as a new temporary destination, the NAF uses the algorithm 2 to determine that the route to new temporary destination is out of the affected area.

Second, we enhance the emergency source services to provide the most up-to-date emergency information. We provide emergency sources with a set of APIs so that they can connect with our framework via web services.

B. NAF Functionality

We describe the functions built into the Navigation Assistance Framework to support the use cases given above. In the sunny day scenario, users can get the navigation assistance from the regular GPS or the ERApp. Without any emergency occurrences during rush hours, users can anticipate their on-time arrivals at their desired destinations. However, emergency incidents do occur at unexpected times and have the potential to create a long delay in travel time. With the ERApp installed on handheld devices, users are able to receive an enhanced CMAS broadcast emergency message [1], [2] that provides

more details about the impending emergency incident. In addition, ERApp is equipped to receive frequent updates about the impending emergency status including tracking information such as GPS location, time, intensity, effected area, etc. This information is necessary for the ERApp to better advise and direct users to their destinations. The goal is to avoid possible road blocks and dangerous areas that are being affected by emergency incidents. We can achieve this goal if we have updated emergency information.

1) Emergency Data: Depending on the type of emergency, information may come from different sources. For example, tornado data may come from the National Weather Center. Hurricane data may be retrieved from the National Hurricane Center. Road closures in the local area may come from the local police department or the Department of Transportation. Therefore, we need to establish a method of retrieving these emergency data from various sources and determine if the connection is either push, pull, or both.

The Navigation Assistance Framework acts as a centralized emergency assistance process which dispatches emergency information to all devices and receives regular updates from emergency sources. With this, the Navigation Assistance Framework needs to subscribe to all of the emergency sources to pull and push emergency data. In addition, the NAF needs to allow ERApp to subscribe in order to receive emergency updates. Depending on users' circumstances, the NAF supports either pull or push subscriptions from ERApp.

Each emergency has its own data set relevant to our framework. For a tornado, we collect the following emergency data: time when the tornado touched down, its track including the GPS location of the tornado, wind speed and direction, and the storm intensity measured in Fujita Scale (F-Scale) [8].

For road closures such as a car accident, road construction, water main break, etc., we collect the following emergency data: starting date and time of the closure, the anticipated date and time of the re-opening of the road, GPS location, and the radius of the affected area. The purpose of getting this data is to provide the magnitude of the emergency, GPS location, and its severity. This allows the framework to approximate the danger area and to provide frequent updates to ERApp so that ERApp can assist users to navigate around the danger area.

2) Updating the Directions: After the Navigation Assistance Framework receives the emergency data from the Emergency Source, it sends the updated data using text messaging to all the devices that have installed ERApp and registered with the NAF. The ERApp determines the next step. ERApp will formulate a new query to get the updated routes to the destination, if the impending emergency is going to be in the forecast path. We make a general assumption that users will manually enter into their event calendars the location addresses of where they will be and from what time to what time they are going to be there. These calendar fields such as location, time start and time end are in the Internet Calendar Specification [11]. The ERApp will use this location address as the destination or allow users to enter their current destinations. The ERApp will send the GPS location and the radius of the affected area to Google Maps, which in turn provides the updated directions to the destination.

The format of the updated text message sent from the NAF to ERApps must be agreed upon and interpreted. The format is a list of name-value pairs. The name must be abbreviated by 2 capitalized characters, where character is one byte, and the value must be a primitive type. These names and and associated types must be accessible by the NAF and the ERApp. Table I provides these names, associated types and short descriptions of each attribute.

TABLE I
TEXT MESSAGE VALUES

Parameter	Type	Description
ID	Integer	Emergency Identification
ET	Integer	Emergency Type
LT	Long	Latitude
LN	Long	Longitude
RD	Integer	Radius
DR	Byte	Emergency Direction
SP	Integer	Speed of moving emergency
AP	String	Advice policies in XML format

According to the SMS Specification [9], an SMS Message has a limitation of 160 readable characters. If we represent this SMS message as readable characters, there may not be enough readable characters to hold values of all these attributes. Therefore, we represent these values as binary values and encode them using the Base 64 Encoding [10] in order to fit within 90 character as described in our previous work [1].

C. Implementation

This subsection will detail the implementation of the Navigation Assistance Framework. We also suggest some interfaces that Emergency Sources and Google Maps Web Services need to implement to make this framework. However, we provide a prototype implementation of these interfaces working together.

1) Emergency Data Collection: Figure 1 shows that the NAF receives emergency data from two sources such as CMAS and Emergency Sources (including periodic updates). CMAS sends the CMAS message [1] to the NAF in its broadcast. Two important pieces of data are the GPS location of the impending emergency and the radius of the affected area. NAF uses this information to load the impending emergency details onto the map as shown in Figure 3. The Emergency Sources can push it to the NAF directly. Table II shows NAF interfaces with the Emergency Sources.

Method	Return Type	Parameters	Description
sendUpdate	int	(int iEID, HashMap mapNVP)	Send update values for the impending emergency.
pullUpdate	HashMap	(int iEID)	Return the update name-value pair hashmap for the impending emergency.

TABLE II
INTERFACE FOR IESOURCE IMPLEMENTATION

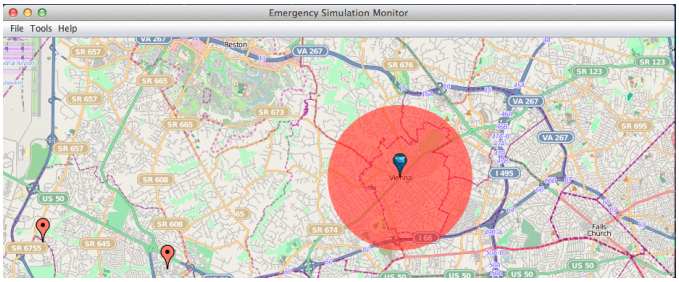


Fig. 3. Impending Tornado

Emergency	Data Name Space	Data Type	Description
Tornado	TND_Longitude	Long	GPS longitude location.
	TND_Latitute	Long	GPS location latitude location.
	TND_Direction	String	Current Direction: East, West, North, South, North-East, South-West, etc.
	TND_Wind_Speed	Integer	Wind Speed measured in miles/hour.
	TND_Radius	Integer	Radius of the affected area measured in mi (miles), m (meters), km (kilometers).
Road Blocks	RB_Longitude	Long	GPS longitude location.
	RB_Latitute	Long	GPS location latitude location.
	RB_Radius	Integer	Radius of the affected area measured in mi (miles), m (meters), km (kilometers).

TABLE III
DATA NAME SPACE

In order for NAF to interpret data, we use the following naming conventions. Table III suggested names and types are the binding agreements between the NAF and Emergency Sources, that we use to retrieve associated values and convert them.

2) Navigation Update: The NAF sends an updated text message to all the registered ERApps installed on mobile devices. ERApps decode the message and extract the emergency information. The ERApp then uses this information to query the Google Maps web services for updated directions. In the prototype implementation, we use Google Calendar to retrieve users' calendar event information such as the location and the time.

Algorithm 1 computes driving directions that avoid the affected area of the impending emergency and helps users navigate to their destinations. The algorithm takes three parameters. The first parameter *mapNVP* is the hash map containing name-value pairs. The second parameter *up* is the user profile, which contains the email credentials to access the calendar. The third parameter *calURL* is the Google Calendar web service URL. On lines 1 and 2, the algorithm initializes two temporary variables *xmlDirDoc* and *calEvent* to null respectively. The *xmlDirDoc* is the updated direction in XML format, which is the return value for this algorithm. On line 3, the calendar service is created for the ERApp client *cCal*. On line 4, client calendar is set with the credentials including the email address and the email passcode, which are used to authenticate the calendar service. The calendar query is created from the calendar URL on line 5. We begin to query calendar events from the calendar service on line 6. We check if there is any entry in the return result on line 7. We then sort all the events based on time from the earliest to the latest on line 8. We create the event iterator on line 9 and go through all the calendar events on line 10. We retrieve the calendar event entry *calEntry* on line 11. If the event time is greater than the current time on line 12, we set the calendar event *calEvent* to *calEntry* on line 13 and exit out of the while loop on line 14.

On line 18, the *calEvent* is tested for null value. If it is null, then the algorithm ends there and return the null *xmlDirDoc* on line 31. If *calEvent* is not null, the destination will be retrieved from the calendar event on line 19. The algorithm retrieves the emergency latitude, longitude, radius of the affected area from the hash map name-value pairs *mapNVP* on lines 20, 21, and 22 respectively. The Algorithm then forms the Google map URL *urlDir* with parameters such as the current location, destination, affected area radius, and the emergency GPS location on line 23. The URL object is created from the *urlDir* on line 24. The input stream *inputStream* is created from the URL object on line 25. On line 26, the document builder factory *dbf* instance is created, which in turn creates

Algorithm 1 :**getUpdatedDirections** Algorithm (Input: HashMap mapNVP, UserProfile up, String calURL)

Require: $mapNVP \neq null$
Require: $up \neq null$
Require: $calURL \neq null$
1: $xmlDirDoc \leftarrow null$
2: $calEvent \leftarrow null$
3: $cCal \leftarrow newCalendarService()$
4: $cCal.setCreds(up.getWEmail(), up.getWEmailPC())$
5: $calQuery \leftarrow newCalendarQuery(calURL)$
6: $resultEvents \leftarrow cCal.query(calQuery)$
7: **if** $resultEvents.getEntries().size() > 0$ **then**
8: $resultEvents \leftarrow sortEvents(resultEvents)$
9: $iterEvents \leftarrow resultEvents.getEntries().iterator()$
10: **while** $iterEvents.hasNext()$ **do**
11: $calEntry \leftarrow iterEvents.next()$
12: **if** $calEntry.getTimeStart() > now()$ **then**
13: $calEvent \leftarrow calEntry$
14: $break$
15: **end if**
16: **end while**
17: **end if**
18: **if** $calEvent\ is\ NOT\ null$ **then**
19: $dest \leftarrow calEvent.getLocation()$
20: $erLat \leftarrow mapNVP.get("LT")$
21: $erLon \leftarrow mapNVP.get("LN")$
22: $erR \leftarrow mapNVP.get("RD")$
23: $urlDir \leftarrow formURL(erLat, erLon, erR, dest)$
24: $url \leftarrow URL(urlDir)$
25: $inputStream \leftarrow url.openstream()$
26: $dbf \leftarrow DocumentBuilderFactory.newInstance()$
27: $db \leftarrow dbf.newDocumentBuilder()$
28: $xmlDirDoc \leftarrow db.parse(inputStream)$
29: $xmlDirDoc.getDocumentElement().normalize()$
30: **end if**
31: **return** $xmlDirDoc$

the document builder *db* on line 27. The algorithm parses the input stream to create the XML document *xmlDirDoc* on line 28. The document element is then normalized on line 29. The algorithm returns the *xmlDirDoc* on line 31. The ERApp can invoke any generic built-in application such as Maps, Navigation, etc. with the *xmlDirDoc* updated direction to provide assistance to the users.

In this algorithm, we only address the immediate event that requires a user's attention and participation during the emergency time. Any calendar events occurring thereafter are not addressed in this paper.

3) Determine the Need for Alternative Routes: This section discusses the algorithm to determine if commuters need to get an alternative route to their destination. If the travel direction of the mobile users to the destination crosses the emergency area determined by the emergency location and its affected area radius, we need to get an alternative route to avoid the emergency area. We can easily retrieve the directions of users' moving vehicles by using the Accelerometer sensor and GPS sensor to determine the vector (speed and direction) of the moving vehicle. But this direction is only temporary and not necessarily the primary direction of where they are heading. Therefore, we need to retrieve the direction from their current position to their destination. We retrieve the location, speed, direction, and the affected area (radius from the emergency location) of the impending emergency from the CMAS message as discussed in section III-C2.

Algorithm 2 : **isAltRouteNeeded** Algorithm (Input: HashMap mapNVP, GPSLocation gpsULoc, GPSLocation gpsDest)

Require: $mapNVP \neq null$
Require: $gpsULoc \neq null$
Require: $gpsDest \neq null$
1: $isAltRouteNeeded \leftarrow false$
2: $lat \leftarrow mapNVP.gets("LT")$
3: $lng \leftarrow mapNVP.gets("LN")$
4: $rd \leftarrow mapNVP.gets("RD")$
5: $gpsELoc \leftarrow new\ GPSLocation(lat,\ lng)$
6: $distUtoE \leftarrow getFlyingDist(gpsULoc,\ gpsELoc)$
7: $bearingEL \leftarrow calculateBearing(gpsULoc,\ gpsELoc)$
8: $bearingDest \leftarrow calculateBearing(gpsULoc,\ gpsDest)$
9: $angleUEtoT \leftarrow arcsin(rd/distUtoE)$
10: $angleBE \leftarrow bearingEL - angleUEtoT$
11: $angleEE \leftarrow bearingEL + angleUEtoT$
12: **if** $bearingDest \geq angleBE\ AND\ bearingDest \leq angleEE$ **then**
13: $isAltRouteNeeded \leftarrow true$
14: **end if**
15: **return** $isAltRouteNeeded$

Algorithm 2 discusses the need for the alternative routes. The algorithm accepts three parameters: *mapNVP*, *gpsULoc*, and *gpsDest*. The first parameter is the hash map of the name-value pairs from the SMS message sent by the NAF. The second parameter is the GPS user location. And the third paramter is the GPS destination location. On line 1, *isAltRouteNeeded* is false. On lines 2 to 4, latitude *lat*, longitude *lng*, and affected area radius *rd* of the emergency are retrieved. The GPS location *gpsELoc* of the emergency is created on line 5. The flying distance [16] *distUtoE* from the user location to the emergency location is calculated on line 6. The bearing angle *bearingEL* formed between the North and the line from the user location to emergency location is calculated one line 7. The bearing angle *bearingDest* formed by the Northern line and the line from the user location to the destination location is calculated on line 8. The angle *angleUEtoT* formed by the line from the user location to the emergency location and the tangent line is calculated on line 9. The angle *angleBE* marked the beginning of the emergency effected area is calculated on line 10. On line 11, the angle *angleEE* marked the end of the emergency effected area is calculated. We are now ready to verify if the bearing angle of the destination is in the angle range of the beginning and end of the emergency affected area on line 12. If the *bearingDest* is within the range, the *isAltRouteNeeded* is set to true on line 13 and returned on line 15. The figure 4 provides the visual map these locations.

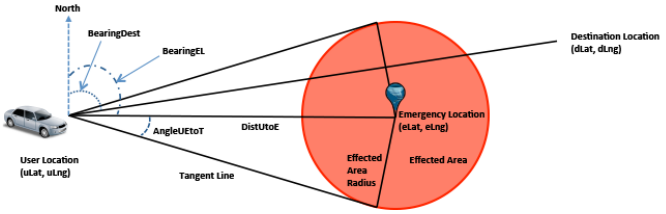


Fig. 4. Alternative Route Decision

4) Providing Emergency Advice: The ERApp uses the accelerometer sensor built in the hand-held devices to detect the user's movement. By comparing the (lat, long) acceleration components, the ERApp can estimate if the user is moving or at rest. The ERApp then compares the distance between the user location and the emergency location to know if he is approaching the emergency area. If the distance calculation indicates that the user is moving toward the emergency area, the ERApp can provide some intelligent advice to the user based on the nature of the emergency.

The advice can also be given based on the user location with respect to the impending emergency. For example, if the user is inside the affected area of a tornado, relevant advice would be to drive to the nearest shelter immediately. ERApp can compare the distance from the user location to the emergency location with the radius of the affected area to see if the user is inside the affected area.

As described in Figure 2, Emergency Sources sent the emergency advice to the NAF in XACML policies. The ERApp applies these policies to see if the user's behavior status satisfy the conditions on the policy. The ERApp displays the recommended advice to the user.

IV. Experimentation

In order to start the experimentation, we need to generate a tornado alert informing the all people in the local area

that the tornado is coming. In our experiment, we set the emergency location to be in Vienna, Virginia, the radius of the effected area to be 3200 meters from the center of the tornado, the expired time, category, certainty, status, urgency, and severity of the tornado. We broadcast the CMAS message to an emulator. Figure 5 shows the preparation of the tornado alert and the ERApp running on the emulator showing the user's location.

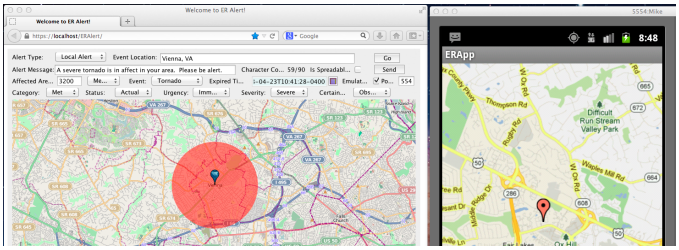


Fig. 5. Prepare the Tornado Alert

The CMAS authority is ready to send the broadcast tornado alert to the emulator by clicking on the Send button. Figure 6 shows the tornado with the effected area in red and the user's location.

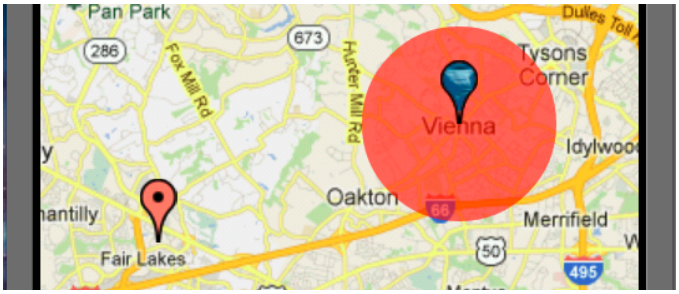


Fig. 6. Tornado Alert on ERApp

We built an Emergency Response Simulation Monitor (ERSimMon) prototype to simulate an individual driving to work during an ongoing tornado. Figure 3 shows the map of the area, the ongoing tornado, and several marker points. These marker points represent users that are currently on the map. There are some configuration settings that are necessary for the simulation. In these configuration settings, Distance Increment is set to 50 meters for the duration of 200 milliseconds, which is indicated by the Sleeping Duration. Emulation Location and ADB Location are required to run the Android Phone emulation. Speed Display is set to Miles per Hour. As the user moves from his location to the destination, it can display the speed at which the user is moving.

In addition, the ERSimMon allows to search, add, modify, or delete markers. Two required fields are the user name and the emulator name. The Address indicates the start point of the user on the map. The Destination Address indicates the ending point of the user on the map after the simulation is complete. These addresses are real address because we use the Google Maps API web service to retrieve the user's location and place the marker on the map. Telnet Server is the loopback server that the emulator is running on. The ERSimMon will set the new position as the user makes a movement. This process simulates the actual driving of the user and at a certain time interval, the GPS on the user hand-held device will detect its new position, which triggers the ERApp to update the position on the map.

The ERSimMon simulates the driving of the selected user and spawns the emulator for that user, showing the user's location and the ongoing tornado. The ERSimMon determines the bearing angle between the user's location to the destination to be 85.64 degrees, the bearing angle between the user's location to the first tangent line to the effected area to be 50.35 degrees and the bearing angle between the user's location to the second tangent line to be 86.73 degrees. Clearly, the user is in the path of the tornado. The ERSimMon presents the alternative route to the user. Figure 7 shows the user driving at the speed of 55.92 miles per hour into the effected area.

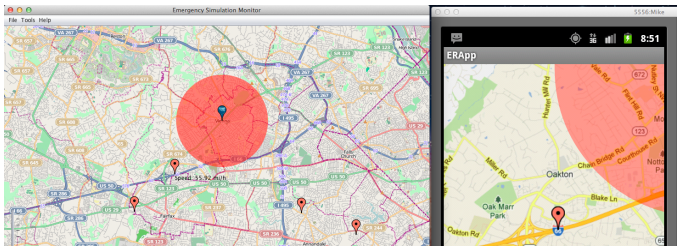


Fig. 7. Driving toward the Affected Area

If we choose to take the alternative route, the user is taking a different route, Route 50 instead of Route 66 with the driving speed of 29.08 miles per hour. Figure 8 shows that the alternative route helps the user avoid the affected area of the ongoing tornado.

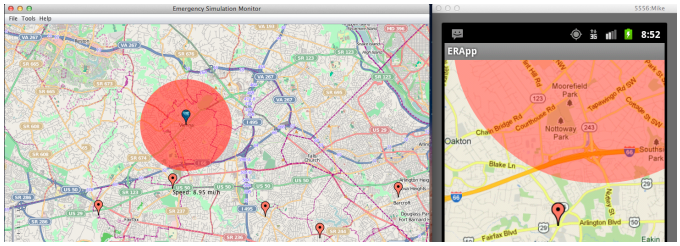


Fig. 8. Avoiding the Affected Area

V. Related Works

This section briefly discusses other significant works aimed at improving or providing the mobile users' navigation assistance.

Although we haven't found any publications that are in this research area, there are other related works on navigation assistance. However, their targets have been for different groups of audiences such as indoor users and blind audiences, one of which is the Guiding Light system [22] that uses projections based augmented reality from the hand-held projectors to

provide way-finding information. This system uses a combination of hand-held sensors such as proximity, accelerometer, compass, and vision to gather and places information on the surrounding spaces. It then compiles all the reference walls, paths, and other stationary objects in its repository. The system then presents the fast-forward clip of the paths and objects that they will encounter when moving from one place to the other in the building.

The second is the General Framework for a Collaborative Mobile Indoor Navigation Assistance System [23]. This system is to provide a cost-effective method to effectively transfer what the user is seeing to a remote expert who is familiar with the area (e.g., providing museum tours, guiding a lost pedestrian, and providing guided emergency response to an area struck by hurricane), such that interactive assistance can be provided to the local user using augmented reality techniques.

Treuillet et al. [24] presents a new approach for localizing a person by using a single-body-mounted camera and computer vision techniques to guide and navigate a blind person within a navigation corridor less than 1 meter wide along the intended path.

Clearly, published works up to now have addressed the indoor navigation assistance or to the specific groups of users. To the best of our knowledge, there has been little or no research done in guiding drivers in emergency conditions.

VI. Conclusions

We have addressed the navigation problem during an emergency by building a Navigation Assistance Framework for Emergencies to provide the navigation assistance to mobile users or commuters. The NAF is collecting emergency data from Emergency Sources and disseminating it to all the registered mobile users. When there is a new update to emergency data, the Emergency Source pushes the new information to the NAF, which in turn updates all of its register ERApps via SMS message. ERApp sends a new query to Google Maps for an alternative route to the destination in order to avoid the emergency path. We also build the ERSimMon to simulate a tornado event and the driving from one place to another to avoid the tornado and its affected area. It also spawns users' emulators in the simulation process to show what is being displayed on the user's ERApp. We suggest interfaces for the Emergency Sources to implement and to send the emergency data to the NAF.

References

[1] Paul Ngo and Duminda Wijesekera, *Emergency Message In CMAS*, In Proceedings of the International Conference on Critial Infrastructure Protection - Sixth IFIP WG, Mar 2012.

[2] Paul Ngo and Duminda Wijesekera, *Enhancing CMAS Usability*, In Proceedings of the International Conference on Critial Infrastructure Protection - Fifth IFIP WG, Mar 2011.

[3] Paul Ngo and Duminda Wijesekera, *Using Ontological Information to Enhance Responder Availability in Emergency Response*, In Proceedings of the Semantic Technology for Intelligence, Defense, and Security Conference - STIDS, 2010.

[4] ATIS-0700006, *CMAS via GSM/UMTS Cell Broadcast Service Specification*; March 2010.

[5] ATIS-0700007, *Implementation Guidelines and Best Practices for GSM/UMTS Cell Broadcast Service Specification*; October 2009.

[6] Commercial Mobile Alert Service Architecture and Requirements. http://www.npstc.org/documents/PMG-0035_Final_Recommendations_v0.6.pdf

[7] The Google Directions API https://developers.google.com/maps/documentation/directions/

[8] Tornado Data http://www.srh.noaa.gov/oun/?n=tornadodata-county

[9] Technical realization of the Short Message Service (SMS) http://www.3gpp.org/ftp/Specs/html-info/23040.htm

[10] Base 64 http://en.wikipedia.org/wiki/Base64

[11] Internet Calendaring and Scheduling Core Object Specification http://tools.ietf.org/html/rfc5545

[12] Out-of-State and Long Commutes: 2011 http://www.census.gov/hhes/commuting/files/2012/ACS-20.pdf

[13] Commuting in the United States: 2009 http://www.census.gov/prod/2011pubs/acs-15.pdf

[14] Poll: Traffic in the United States http://abcnews.go.com/Technology/Traffic/story?id=485098&page=2#.UVDUaBkbqL8

[15] Commercial Mobile Alert System (CMAS) http://www.fema.gov/commercial-mobile-alert-system

[16] Calculate distance, bearing and more between Latitude/Longitude points http://www.movable-type.co.uk/scripts/latlong.html

[17] XACML. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

[18] National Hurricane Center. http://www.nhc.noaa.gov/

[19] National Weather Center. http://nwc.ou.edu/

[20] Dynamic Mobile Application. http://www.its.dot.gov/dma/index.htm

[21] Imagine... http://www.its.dot.gov/imagine.htm#two

[22] J. Chung, I. Kim, and C. Schmandt, Guiding light: navigation assistance system using projection based augmented reality, in Proceedings of the IEEE International Conference on Consumer Electronics (ICCE11), IEEE, 2011, pp. 881882, doi: 10.1109/ICCE.2011.5722917.

[23] Rao, H. and Fu, W.-T. "A General Framework for a Collaborative Mobile Indoor Navigation Assistance System." In Proceedings of the ACM International Conference on Intelligent User Interfaces (IUI), Santa Monica, CA. 2013.

[24] S. Treuillet and E. Royer, "Outdoor/Indoor Vision-Based Localization For Blind Pedestrian Navigation Assistance", International Journal of Image and Graphics Vol. 10, No. 4 (2010) 481496. DOI: 10.1142/S0219467810003937

[25] Shortest Path http://www.cs.princeton.edu/~rs/AlgsDS07/15ShortestPaths.pdf

Position Papers

Big Data for Combating Cyber Attacks

Terry Janssen, PhD, SAIC
Chief Scientist & Cyber Strategist
Cyber Operations
Washington, D.C. USA
terry.l.janssen@saic.com

Nancy Grady, PhD, SAIC
Technical Fellow, Data Science
Emerging Technologies
Oak Ridge, TN USA
nancy.w.grady@saic.com

***Abstract*—This position paper explores a means of improving cybersecurity using Big Data technologies augmented by ontology for preventing or reducing losses from cyber attacks. Because of the priority of this threat to national security, it is necessary to attain results far superior to those found in modern-day security operations centers (SOCs). Focus is on the potential application of ontology engineering to this end. Issues and potential next steps are discussed.**

Keywords—big data; ontology; cybersecurity; modeling, search; discovery; analytics; variety; metadata

I. INTRODUCTION

The last few years have seen tremendous increases in the amount of data being generated and used to provide capabilities never before possible. "Big Data" refers to the new engineering paradigm that scales data systems horizontally to use a collection of distributed resources, rather than only the earlier vertical scaling that brought faster processors and more data storage into a single monolithic data platform. Big Data technologies have the potential to revolutionize our capabilities to handle the large datasets generated in any cyber data analytics. The challenge, however, is not just in handling the large volumes and high data generation rate, but in leveraging all available data sources to provide better and faster analytics for attack detection and response. In this paper, we will discuss Big Data analytics, metadata, and semantics for data integration, and applications to cybersecurity and cyber data management.

II. BIG DATA

Big Data has several defining characteristics, including ***volume***, ***variety*** (of data types and domains-of-origin), and the data flow characteristics of ***velocity*** (rate) and ***variability*** (change in rate) in which the data is generated and collected.

Traditional data systems collect data and curate it into information stored in a data warehouse, with a schema tuned for the specific analytics for which the data warehouse was built. *Velocity* refers to a characteristic that has been previously referred to as streaming data. The log data from cell phones, for example, flows rapidly into systems, and alerting and analytics are done on the fly before the curation and routing of data or aggregated information into persistent storage. In a Big Data architecture, this implies the addition of application servers to handle the load. *Variability* refers to changes in the data flow's velocity, which for cost-effectiveness leads to the automated spawning of additional processors in cloud systems to handle the load as it increases, and release the resources as the load diminishes. *Volume* is the dataset characteristic most identified with Big Data. The engineering revolution began due to the massive datasets from web and system logs. The implication has been the storage of the data in its raw format, onto distributed resources, with the curation and imposition of a schema only when the data is read.

Big Data Analytics. Much of the development of Big Data engineering is a result of the need to analyze massive web log data. Massive web logs were first filtered by page for aggregate page counts, to determine the popularity of pages. Then the pages were analyzed for sessions (spawning the now massive "cookie" industry to make this simpler). "Sessions" are the sequence of activities that describe a customer's interaction with the site at a "single-setting," with the analyst describing what time-window is considered a session. The next step in analytics capability came from the realization that these sessions could be abstracted into patterns rather than being treated as just the literal collection of pages. With this step, traversal patterns helped site designers see the efficiencies in their link structure. Furthermore, these usage patterns could in some cases be attached to a customer account record. With this step, the site could be tuned to benefit the most valuable customers, with separate paths being designed for the casual visitor to browse, leaving the easy efficient handling for loyal customers. This pattern-oriented analysis applies to the cyber domain, in analyzing logs from a server.

The last 15 years have seen the extension of a number of analytics techniques to leverage the horizontal Big Data scaling paradigm to address both log and linked-node data found in social sites. The cyber community can leverage web log and Social Network Analysis to use the massive amounts of data to determine session patterns and the appropriateness of activity between resources. The challenge is that cyber must also deal with a richer set of attributes for the resources and their expected/allowed interconnections, which adds in a variety of other contextual datasets into the analysis.

Variety. Traditional systems handled the variety of data through a laborious integration process to standardize terminology, normalize into relational tables, choose indexes, and store into a data warehouse that is tuned for the specific analytics that are needed. This is an inflexible process that does not easily accommodate new data sources, changes into underlying data feeds, or new analytical requirements.

For web log analysis, this extension to customer session analytics only required the assignment of a customer or visitor

ID to the session, allowing integration with a purchasing history. In the cyber analytics case, the integration point is not so simple. The integration of packet data, with server log data, with port-to-port connectivity data, with server type data, with network router settings, and so forth, provides a more complex use case, needing a more sophisticated way to integrate such a variety of data, some of which carries a number of additional attributes that are needed.

Recently, *variety* datasets have been addressed through mashups that dynamically integrated a couple of datasets from multiple domains to provide new business capabilities. Early mashups demonstrated this value, for example, in the integration of crime data with real estate listings; a valuable analysis that was not possible before the availability of open datasets. There is a limitation to such mashups because of the integration of a limited number of datasets, with the integration variables being manually selected. This type of manual integration is insufficient for analytics across different large volume datasets with complex inter-relationships.

Variety is the Big Data attribute that will enable more sophisticated cyber analytics. The requirement is for an automated mechanism to integrate multiple highly diverse datasets in an automated and scalable way. This is best achieved through a controlled metadata.

III. METADATA

The executive branch has been pushing an open data initiative to move the federal government into being a data steward. The goal in releasing the data is to better serve the public and promote economic growth through the reuse of this data. The difficulty in using this data arises from the lack of the metadata descriptions. Data reuse requires as much information as possible on the *provenance* of data; the full history of the methods used for collection, curation, and analysis. Proper metadata increases the chances that datasets are re-purposed correctly—leading to analytical conclusions that are less likely to be flawed.

Two mechanisms are used for dataset integration in a relational model. In the relational model, lookup tables are established to translate to a common vocabulary for views, and a one-to-one correspondence is used to create keys between tables. In a NoSQL environment, joins are not possible so table lookups and or keys cannot be used for data integration. The connection of data across datasets must reside in the query logic and must rely on information external to the datasets. This metadata logic must be used to select the relevant data for later integration and analysis, implying the need for both standard representation and additional attributes to achieve the automated data retrieval.

A second approach is used to speed the data integration process for manual mashups of diverse datasets. Often XML wrappers are used to encapsulate the data elements, with the nomenclature for each dataset provided in the wrapper, based on user interpretation of the data elements. This approach allows rapid integration of data through the wrappers (as opposed to a lengthy data warehouse integration), but it is not an approach that can be automated, nor can it be used for large volume datasets that cannot be copied due to their volume. Even in a mashup, wrapper terms used in the metadata are themselves subject to interpretation, making reuse of data elements difficult.

Without metadata referenced to well-understood standard terminology applicable across domains, the diverse datasets cannot be integrated automatically. In addition, the integrating elements must be applied outside the big data storage, implying that the integration logic must reside in the metadata layer.

IV. SEMANTIC TECHNOLOGY

Semantic technologies are crucial for the future handling of big datasets across multiple domains. While we have methods for unique concept identification arising through the Semantic Web, these technologies have not made inroads into traditional data management systems. Traditionally, the ETL process has been used to enforce standard terminology across datasets, with foreign keys to external tables for the related information. This is not a scalable solution, since the introduction of a new data source requires the careful construction of foreign keys to each other dataset in the database. This lack of extensibility to add in additional sources highlights the limitations of horizontal scalability in current approaches. In addition, there are limitations on the continued expansion in large data warehouses, highlighting their inability to continue to scale vertically.

Semantic technologies have not yet made inroads into Big Data systems. Big datasets that consist of *volume* tend to be monolithic with no integration across datasets. The data is typically stored in its raw state (as generated), and no joins were allowed in the initial Big Data engineering. Given this, most Big Data analytics approaches apply to single datasets.

For solutions addressing the integration of *variety* datasets, the ability to integrate the datasets with uniquely defining semantic technology is a fundamental requirement. Two overarching requirements need to be addressed to use ontology for the integration of Big Data: constructing the ontology and using the ontology to integrate big datasets.

Ontology scaling. The standard method for data access through an ontology is to ingest the data into an ontological database, where the data elements are encoded along with their extant relationships. This does not work in a Big Data scenario, since ontological databases do not have the horizontal scalability needed to handle data at high volume, velocity, or diversity. Further exacerbating the problem is that some of the data needing to be integrated are not owned by the analytical organization and cannot be ingested, but only accessed through query subsets.

Separate ontology for metadata. The implementation of an integrating ontology would consequently need to reside in the metadata for browsing and querying. While this metadata could be browsed manually, the real value comes if it can be actionable; such that selections over the metadata ontology would automatically construct queries to the Big Data repository. A number of ontologies relative to the cyber domain already exist, encompassing resources, attack event ontologies, and so forth. The key is to incorporate the appropriate elements and their relationships needed to describe the elements in the desired datasets. Our intent is not to recreate a cyber ontology from scratch, but to leverage those that exist to develop a first order ontology specific to the integration of the relevant cyber datasets. Focusing on first order logic will enable the ontology to be actionable to dynamic data integration.

In order to serve as the facilitator for data integration for automated integration, this first order ontology would need to contain elements such as: data element definitions, dataset location, data producing resource characteristics, and resource connectivity.

For analytics, additional mid-level ontologies would be needed to provide reasoning over the data, such as time and location. Domain-specific ontology elements would include, for example, resource attributes by resource type, translations such as Internet protocol (IP) to location, and derived attack pattern components.

The key to the use of a semantic representation for the metadata is separating the semantic metadata from the data storage. In order to leverage the scalability and speed of high-volume NoSQL solutions, the ontology will need to reside in its own scalable environment. Data exploration would require a mechanism to browse the metadata within the ontology, with a seamless transfer mechanism to flow down into the data.

Probabilistic Challenges. One significant challenge in the use of ontology for automated data analytics across datasets resides in the need for probabilistic reasoning. Typically in ontology representations, triplets are considered *"facts,"* implying full confidence in the data elements being described. In the real world, such a luxury is typically non-existent. Resources will continually be updated, and there will be latency before the new configurations are updated in the ontology. Attack chains will have multiple possible paths with probabilistic representations of each link type. Activity counts must be evaluated with a statistical significance test to determine if an activity is truly of concern. Such counts will have variations relative to time of day and day of week. Using an ontology for such probabilistic analytics will require the ability to analyze activity under some uncertainty. Much work has been done on probabilistic ontology, like MEBN, which inserts Bayes' theorem in ontology nodes [1].

V. APPLICATION TO CYBERSECURITY

Practical application to countering cyber attack is achievable in the near-term. The following questions can be answered with properly implemented Big Data technologies that span the variety of datasets: What data is available on malware X attacks globally? How many machines did an event land on? What ports were leveraged? What users were affected? What machines were compromised? What was leaked? Was sensitive information lost? Who did it? Was it an insider or outsider? More difficult questions for the future would be: What is the composite activity globally of this attacker that penetration tested (pentested) my perimeter? What are all the locations globally of <malware name> attacks? What should I expect from this attacker within the next hour? Next week? Next month? (Based-on the historical data on this attacker.) What unsafe actions are my users doing, rank ordered by risk significance? What suspicious activity occurred today? Where is the greatest risk within the enterprise? It would also be useful to tabulate statistics on vulnerabilities versus attacks, and visualize the results.

The latter "future set" of questions requires more research and development in topics like machine learning and reasoning, and is well beyond this paper's scope. For example, can ontology as proposed in this paper help us reason about risk based on the topology of devices and controls? Theoretically, this is deterministic and machines should be able to do better than man. Our intent is to model perimeter security of a large, enterprise network and collect real-time data, reason about risk in real-time based on the topology of devices and controls, and respond to threats in attempt to prevent loss. Given the appropriate set of data and generation of a set of reasonable hypotheses, can we use Big Data to do evidence collection to support or refute those security risk and threat hypotheses, in time to prevent loss?

Progress-to-Date. As a first step in preparing to instantiate an ontology, we have been mindful of what hundreds of organizations do in the current cybersecurity management process in a global networked enterprise. Description of this workflow is beyond this paper's scope. System awareness currently resides in the minds of hundreds of professionals who track threats and malware, maintain the security devices like firewalls and the configurations and patches of thousands of network devices, monitor events and log files, create tickets when an anomaly is observed, and perform remedial actions such as Incident Response; Configuration Management; Vulnerability and Patch Management; Firewall, Intrusion Detection and Prevention; Deep Packet Inspection and Cyber Threat Assessment; Security Architecture and Design; and so forth.

We propose to elicit all knowledge necessary assessment, decision, planning, and response into this ontology. At first glance, this may appear daunting, but based on the successes with ontology engineering in recent years, and the high stakes,

we believe this not only practical, but necessary, to better understand how to solve this national priority problem.

Cyber-security management has the characteristics of a successful knowledge elicitation and ontology engineering endeavor. The information is in digital form, and cyber-security processes are repetitive—meaning that the same indications of an attack are well documented and observed in typical network operations routinely and the remedial steps are documented and used routinely. This is not to say the cybersecurity experts are not highly knowledgeable and skilled—just the opposite. This knowledge can be coded and reused in the parts the machine does best; man should continue to do the parts that it does better than machines. With this expectation, we will meet the goal stated up front of flipping the current situation to one where a network's defense is optimized and efficient, lowering cost of defense, and making it very hard and expensive for the attacker.

Cyber Ontology for Countering Attacks. The top levels are illustrated in Figures 1 and 2.

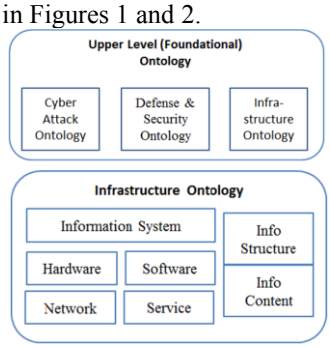


Figure 1. Upper Level and Lower Level Infrastructure Ontology.

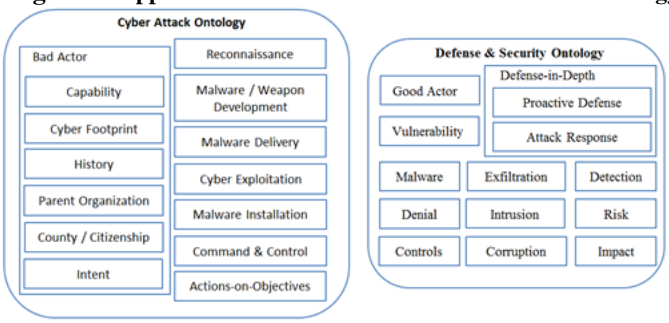


Figure 2. Lower Level Ontology for Attack and Defense.

Our goal is a proof-of-concept prototype of the entire process, but only for a few appropriate types of attacks and respective plans as defined by a fairly rigorous test set. Big Data elements for proof-of-concept have been partially selected.

Ontology engineering tools are being evaluated for "most suitable" for implementing this ontology for use in the system as previously described. A trade study will need to be conducted, for tools that can be selected for implantation of a production system capable of meeting the aforementioned objectives in a large, global enterprise network. For the purpose of demonstrating the concept we selected an ontology engineering tool from highfleet.com that reportedly provides an implementation of first order logic that is decidable and tractable (by simple programming constraint). It is a tool that one of the authors has used in the past. Results here are positive from the little done to-date; we cannot do an assessment until the ontology is populated. There are other ontology engineering tools, for example the description logic Protégé ontology editor. We have not made a decision; eventually we will need to identify appropriate metrics and conduct assessments to determine what would be needed for production grade deployment to address this problem space

Due to page limit constraints, it is impossible to discuss all aspects of the cyber ontology development, but a few aspects need to be mentioned. For example, there are many good resources for specifying and instantiations these ontologies to a level useful in cyber, most notable are efforts by MITRE [2]. Research issues remain unanswered and they can be categorized into big data and analytics, ontology and probabilistic reasoning, decision making and design and architecture. Cybersecurity is a hard problem and it is doubtful that the approach taken in this paper, or any other, will be a complete solution. Furthermore, the cyber attack sophistication is advancing rapidly which compounds this problem significantly [3].

VI. Future Steps

We are in the planning phase for continued research and development, beginning with the Big Data analytics necessary to more fully identify, understand, and respond to cyber attacks. In parallel, we would like to develop a proof-of-concept prototype to test how well this ontology and Big Data integration would work in practice in a large enterprised network with high traffic and large number of cyber attacks. The key to the success of this prototype will be to focus on one narrow aspect of cyber attack defense; if one is implemented and demonstrated, it can be used to extrapolate the resources needed for development and implementation in large production environments.

References

[1] Laskey, K.B, *MEBN: A Language for First-Order Bayesian Knowledge Bases*, Department of Systems Engineering and Operations Research, George Mason University, Fairfax, VA, 2007

[2] Obrst, L., Chase, P., Markeloff, R., Developing an Ontology of the Cyber-security Domain, *Semantic Technology for Intelligence, Defense and Security* (STIDS) 2012, GMU, Fairfax, VA, 2012.

[3] http://www.cnas.org/technology-and-national-security

Hierarchical Decision Making

Matthew J. Lewis
Data Exploitation Systems
Michigan Aerospace Corporation
Ann Arbor, MI
mlewis@michaero.com

***Abstract*—Decision making must be made within an appropriate context; we contend that such context is best represented by a hierarchy of states. The lowest levels of this hierarchy represent the observed raw data, or specific low-level behaviors and decisions. As we ascend the hierarchy, the states become increasingly abstract, representing higher order tactics, strategies, and over-arching mission goals.**

By representing the hierarchy using probabilistic graphical models, we can readily learn the structure and parameters that define a user's behavior by observing his activities over time—what data they use, how it is visualized, and what decisions are made. Once learned, the resulting mathematical models may be combined with the techniques of reinforcement learning to predict behavior and anticipate the needs of the user, delivering appropriate data, visualizations, and recommending optimal actions.

Keywords—decision making; hierarchical hidden Markov models; reinforcement learning.

I. INTRODUCTION

Human operators, particularly in the context of military operations, must quickly make critical decisions. Although these operators have access to unprecedented volumes of diverse data sources involving media reports, financial information, imagery, signals intelligence, and human intelligence, making decisions based on such data is confounded by many factors, including: 1) Limited human and computational resources; 2) difficulty synthesizing a coherent picture from volumes of manifold and (often) irrelevant data; 3) an inability to derive meaning from or detect structure in high dimensional data; and, 4) the randomness and uncertainty intrinsic to the real world. When a human operator is making a decision, much of this data is irrelevant and, worse, confusing.

To reduce the cognitive load of human decision makers and improve the quality of the decisions they make, we can develop frameworks—algorithms, APIs, and user interfaces—that detect what data is relevant and how it should be presented in a manner that is particular to the *decision context*. For our purposes here, a *decision context* specifies: 1) the goal, task, or mission relevant to our decision; 2) the reason the task is important (*i.e.*, a global perspective); 3) any constraints and utilities associated with the decision; 4) previous decisions that were made, as well as likely future ones; and, importantly, 5) the set of candidate decisions available to the decision maker.

In fact, human operators may not explicitly conceive of this decision context, but it nevertheless serves as a useful set of *latent* variables, which help us intelligently aggregate relevant data and determine how to best present it to the human user. We may use machine learning techniques to identify and classify decision contexts, as well as predict which data, in what formats and with what visual representations are most useful.

In this paper we advocate an approach to building flexible frameworks that identify a *decision context* that may help anticipate the types of data the user will find useful, as well as how the data should be represented and visualized. As users interact with the framework to make decisions, entering search terms, selecting data, interacting with tables and plots, and ultimately making decisions, the framework learns and adapts, improving its predictive capabilities and honing its notion of what constitutes a decision context.

II. DECISION CONTEXT

A. The Importance of a Decision Context

How can we learn the decision context for a particular task or goal? We have, at the lowest level, the measured input associated with how the user is using a system to help make a decision. This can be much more than the keywords associated with a database search. It could include what elements of an interface the user clicks on, heat maps of cursor positions, *how long* a given data source is investigated, *what data* elements a user expands—even, if available via camera interfaces, what information the operator is actually looking at, and for how long. The details we record about how an operator interacts with a user interface (UI)—or, even better, how an entire population of operators interacts with a UI—will yield valuable, predictive insight into what data is useful and what data is quickly discarded by the operator, with respect to the estimated decision context. This information can be gleaned from nearly any system by using external monitoring software packages.

B. Decision Context as a Hierarchy

Making the leap from crude interaction measurements to understanding the intent of the human operator and the decisions he is trying to make is difficult—indeed, a problem at the core of machine learning and artificial intelligence.

One approach to attacking this problem is to understand the decision context as a hierarchy, wherein the lowest layers of the hierarchy represent the raw input signals—the measurements of operator interaction with the UI, and the structure and content of requested data; as we move up in the hierarchy, inputs from the lower layers are mapped to increasingly more abstract ideas—operator intent, operator

confusion, mission goal, *etc.*, which together form the decision context. One tool for efficiently representing such hierarchies is the hierarchical hidden Markov model (HHMM) [1].

C. An Example

To take a concrete example, imagine driving a car. At the very lowest level, a driver is taking in visual and auditory information about obstacles and other cars on the road, and making numerous low-level decisions—*press the gas*, *press the brake, turn left*, and so forth. But these decisions are always being made in the context of a higher-order goal—say, navigating to the grocery store. And that goal, in turn, is made in the context of a yet higher order goal—needing to eat. Decisions are made at each level in the hierarchy, and influence the decisions made at the other levels.

Consider Fig. 1, which illustrates a portion of a highly simplified hierarchical hidden Markov model. Each blue node in the network represents a state of the system. These states are arranged into a hierarchy of levels. Suppose you are hungry, so that at the highest level in this hierarchy you are in the *eat* state. There is some probability you will transition to another state at this level — the *sleep* state, for example. However, more likely, because of your hunger, you will transition to a lower level in the hierarchy, perhaps to the node that represents the *go to the grocery store* state. This in turn transitions to yet a lower level, to represent increasingly specific sequences of behavior — *leave the house* followed by *drive to the store*. When behavior at one level of the hierarchy is completed (which happens when one transitions to a black node), control is returned to one level higher in the hierarchy, where it left off. Note that, in general, states transition at lower levels change much more quickly than they do at higher levels: things are changing quickly as we drive down the street, stopping at stop signs, taking right turns, etc. But all the while we are still in the *eat* state—the higher order context for our behaviors has not changed.

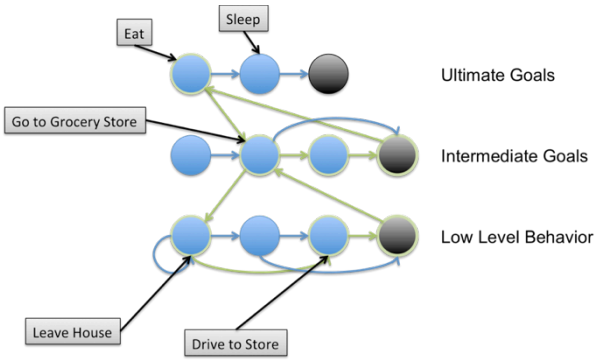


Fig 1: An example of a hierarchical hidden Markov network.

Learning the structure of such a network, as well as the probabilities associated with transitions between states and levels, can be done efficiently and in an unsupervised manner using the mathematics of probabilistic graphical models [2, 3]. We have implemented these techniques in the context of autonomous vehicle control, predictive analytics, and electronic warfare.

Using these mathematical models, we may take low level behavioral inputs and infer the higher order goals that are likely driving this behavior. Conversely, given a higher order goal, we can estimate the behaviors that will likely be used in the context of that goal. This information can be used to optimally configure a user interface, retrieve relevant data, and otherwise support operator decision making. We describe a way to achieve this optimization below.

III. Learning By Interacting with the Environment

A. From States to Actions

We have argued that describing decision context as a hierarchy provides a rich way to describe operator behavior. In a sense, it provides a flexible way of modeling the *state* of the operator — why an operator is doing something, how he is trying to accomplish it, at a strategic level, and what resources he likely needs to support the effort. This representation can determine, at each time step, what the most likely decision an operator is likely to make, based on past behavior. But this representation alone does not provide a mechansim for learning — at a given timestep, the best possible decision.

Consider again the car driving example: if we have an HHMM running, a few observations (our operator is in the car, he is driving toward the grocery store) will quickly establish the decision context (the operator needs to *eat*), and can predict that he will turn left at the upcoming intersection, because he has done so previously, and because it ultimately leads to the grocery store. But it predicts or suggests this decision only because of what has been observed in the past, not because turning left happens to be the fastest way to reach the grocery store. In order to optimize the decision making process, we harness another piece of mathematical technology, known as Reinforcement Learning.

B. Reinforcement Learning

Reinforcement Learning (RL) is a machine learning technique inspired by behavioral psychology, developed to emulate the manner in which humans learn via experience [4]. RL is concerned with teaching an agent how to interact with an environment in order to maximize a *cumulative* reward. RL has been successfully applied to problems across many domains, including industrial planning, autonomous vehicle control, pattern recognition, dynamic channel allocation in the cellular industry, and even the game of chess.

At each time step t of an RL algorithm, the agent finds itself in a state, $s_t \in S$. When in this state, it has available a number of available actions, or decisions, a_t. It selects an action according to a policy, $\pi(s,a)$, which records the probability of selecting action a given the agent is in state s, i.e., $\pi(s,a) = P(a|s)$. Because of this action, the agent transitions to a new state s_{t+1} and receives a scalar reward r_{t+1}.

The reward function is a crucial component. It may be a complex, time dependent function of the state of the agent and his environment; it is used to encode the goals of a decision making task—that is, by optimizing this function, we achieve the planning goal. Surprisingly complex behavior can emerge from the use of even simple scalar reward functions. In this

instance, the reward function could be the utility of the final decision, the speed at which the decision is made, or a combination of these and other measures.

But—and this is critical—the goal of RL is not to maximize the reward received at the very next time step, but rather the total cumulative reward over all time, which we call the *return*:

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots \quad (1)$$

The scalar $\gamma \in [0,1]$ determines the importance of future rewards relative to near-term rewards. If $\gamma = 1$, distant future rewards are as important as near term rewards. For $0 < \gamma < 1$, we refer to R_t as the *discounted return.*

The advantage of this approach is that the reinforcement learning agents are not required to act to maximize short-term gains, but rather learn to act in complex ways to achieve objectives, even if *they must make occasionally suboptimal decisions*.

An object of fundamental interest in RL is the action-value function, which we denote by $Q(s,a)$. The action-value function records the expected discounted return:

$$Q^{\pi}(s,a) = E\{R_t | s_t = s, a_t = a\} \quad (2)$$

The superscript π indicates that the action-value function is relative to the policy π. It tells us the expected value of being in state s and taking action a. The goal of an RL agent is to learn this function by interacting with its environment. Once this function has been learned, determining an optimal policy, π, is straightforward: given we are in state s, we select the action a so that we maximize the expected value—that is, we select the optimal action $a^* = \max_a Q(s,a)$. In fact, we do not always select the optimal action, but sometimes (with probability β) select a suboptimal action. In this way, we manage to avoid the local minima in our reward functions, and may more rapidly adapt as our environment evolves in time.

We learn the Q function iteratively. We begin with an arbitrarily initialized function, $Q(s,a)$. Starting at time t, in state s_t, we select an action a using a policy derived from the function Q—i.e., we select the action with the highest expected value. Because of the action, we find ourselves in state s_{t+1}, and receive reward r_{t+1}. We update the action value function as follows by replacing the value of $Q(s_t, a_t)$ with,

$$Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (3)$$

The scalar $\alpha \in [0,1]$ is a learning rate, which specifies how quickly the system adapts.

After the update, we select another action, and the cycle continues. We can demonstrate (via practical applications and formal mathematical proofs) that this iterative procedure converges to the correct value for the action-value function Q. The policy π is implicit in the action-value function, as emphasized above: when we are in state s we select $a^* = \max_a Q(s,a)$.

Importantly, the RL algorithm does not stop learning after this convergence, for the simple reason that the environment may be changing, and our the agent's behavior may need to adapt accordingly. This occurs naturally in the context of the RL algorithms because there is some nonzero probability (β, as defined above) that we will select a non-optimal action. This ensures that we are trying new things, and although we may not always be making the optimal decision, we can avoid making decidedly poor decisions because our environment has changed from beneath us — there is always, of course, a tradeoff between achieving optimal behavior and responding quickly to a changing environment (i.e., stability vs. maneuverability), but with these methods we can parameterize and quantify the tradeoff.

C. Bringing the Pieces Together

We have discussed two distinct pieces of mathematical equipment that may be used to deal with hierarchical decision making. For the first piece, an HHMM is used to learn the decision contexts relevant to a problem. These decision contexs define a set of *states*. Given a state, and a set of actions that the operator (or the computer) may take, our second piece of technology, Reinforcement Learning, sets out to learn what decisions will lead to the best outcomes, with respect to a set of reward functions.

The fact that these two pieces share a common language—they understand they operator's state and the actions he may take, indicate they may work together effectively. Given the HHMM predicts we're in a particular state, the RL algorithms recommend an action. As a results we transition to a new state, which is estimated by the HHMM, another action is recommended, and the cycle repeats. A human operator may either execute the decision recommended by the RL algorithms, or select another action—and the system learns from the resulting state in either case.

To continue the analogy with the car: As the human operator approaches an intersection, the RL algorithm may understand that the user is in an *eat* state, attempting to go to the grocery store. Typically, at this point, according to the HHMM, the user turns left, but the RL algorithm may recommend right. As a result, the user arrives at the grocery store four minutes faster than usual. This reinforces the RL algorithm's decision to recommend that action, and in the future, it will preferentially recommend it. If at some point something, say construction, renders that driving route unmanageable, the RL algorithms will adapt accordingly, and perhaps again recommend taking a left hand turn at the intersection, instead.

IV. Conclusions

Decision making must be made within the appropriate context, and we contend that *context* is best represented by a hierarchy of states. The lowest levels of this hierarchy represent the observed raw data, or specific low-level behaviors and decisions. As we ascend the hierarchy, the states become increasingly abstract, representing higher order tactics, strategies, and over-arching mission goals.

By representing this hierarchy using probabilistic graphical models, we can readily learn the structure and parameters of a user's behavior by simply observing their activities over time—what data they use, what plots they make and use, etc. Once learned, the resulting mathematical models may be used to intelligently predict behavior, anticipate the needs of the

user, and deliver the appropriate data, visualizations, and other resources before the user even knows he wants it.

Furthermore, given this hierarchical representation, we can use the mathematics of reinforcement learning to help the user make the best possible decision (or decisions) with respect to the specified reward functions.

A. Moving Forward

Although Reinforcement Learning methods have been successfully integrated with probabilistic graphical networks, which allow us to build autonomous decision making systems that learn and adapt from experience, and though these technologies have been applied to a number of disparate fields, including autonomous vehicle control, and electronic warfare, more research needs to be done to develop these into a general framework.

In order to produce a flexible framework, we must create a method for easily defining reward functions in terms of the ultimate decision goals of the system, as well as methods for labeling the states of the decision context hierarchy. In addition, for specific uses, we must develop a consistent method for ingesting data so that it may be readily used by these algorithms.

These and other challenges represent future research efforts in this area.

REFERENCES

[1] S. Fine, Y. Singer and N. Tishby, "The Hierarchical Hidden Markov Model: Analysis and Applications", Machine Learning, vol. 32, p. 41–62, 1998

[2] K. Murphy and M. Paskin. "Linear Time Inference in Hierarchical HMMs", NIPS-01 (Neural Info. Proc. Systems).

[3] K. Murphy, "Dynamic Bayesian Networks: Representation, Inference and Learning", UC Berkeley, Computer Science Division, July 2002.

[4] R. Sutton & A. Barto, “Reinforcement Learning: An Introduction”, MIT Press*, Cambridge, MA, 1998.*

Towards Context-Aware, Real Time and Autonomous Decision Making Using Information Aggregation and Network Analytics

Prithiviraj Dasgupta
Department of Computer Science
University of Nebraska at Omaha
Omaha, Nebraska, 68182-0500
Email: pdasgupta@unomaha.edu

Sanjukta Bhowmick
Department of Computer Science
University of Nebraska at Omaha
Omaha, Nebraska, 68182-0500
Email: sbhowmick@unomaha.edu

***Abstract*—We consider the problem of real-time, proactive decision making for dynamic and time-critical decision-events where the choices made for multiple, individual decisions over time determine the final decision outcome of an event. We posit that the quality of such individual decisions can be significantly improved if human decision makers are provided with decision aids in the form of dynamically updated information and dependencies between the different decision variables, and the humans affecting those decision variables. In this position paper, we propose the CONRAD (CONtext aware Real-time Adaptive Decision making) system that uses computational techniques from large scale network analysis and game theory-based distributed information aggregation to develop such decision aids. CONRAD's functionalities are implemented through three subsystems - a decision making subsystem that updates and mathematically combines information from different decision variables to predict the outcome of the decision event, a decision assessment subsystem that uses the currently predicted decision outcome to estimate the future decision trajectory and recommends information collection-related actions to the human decision maker, and, a network analysis subsystem that uses those recommended actions to dynamically update the dependencies and correlations between events and people influencing the decision variables. To the best of our knowledge, our work is one of the first attempts towards combining dynamic decision updates and using the predicted decision trajectory as a proactive feedback mechanism to dynamically update the correlations between decision variables so that human decision makers can make more strategically-informed and well-aligned decisions towards the desired outcome of decision events.**

I. Introduction

Modern decision making scenarios are characterized by large amounts of data and information that arrive dynamically, over a short period of time, from multiple sources. Processing this data in a time-critical manner to make accurate decisions is an overwhelming task for human decision makers. Over the past few decades several decision making solutions have been proposed to aid human decision makers with tools such as intelligent or automated software that use computational methods and mathematical models of human cognitive processes to make sophisticated decisions on behalf of humans[3], [12]. However, most existing decision support tools provide only limited context awareness of the decision process to the decision maker in rapidly evolving, information-rich and time-critical scenarios. This reduces the efficiency of human decision makers in making accurate decisions, and, consequently, could result in erroneous decision making in critical situations. Therefore, it makes sense to investigate techniques that could alleviate the human decision makers' context awareness by presenting information relevant to the decision making process, precisely and in a timely manner, to the decision maker.

To address this problem, we present the framework of a context-aware, real-time, decision making system called the CONRAD (CONtext aware Real-time Adaptive Decision making) system that focuses on enabling and enhancing the capabilities of human decision makers by developing proactive decision-aids for making high-accuracy, time-critical decisions in complex, data- and information-rich environments. The central research problem that CONRAD proposes to answer is as follows: *Given a set of decision variables in the current decision context along with a set of data sources from which the decision variables can be derived and/or calculated and updated, what is a suitable set of techniques for (i) extracting relevant information, (ii) then using that information to update, correlate and aggregate the decision variables dynamically, and finally , (iii) assessing the quality of the aggregated decision outcome (prediction), so that the divergence between the aggregated decision outcome (predictions) and the desired decision outcome is successively minimized?*

To address this question, we propose to represent a decision event as a collection of decision variables that are affected by the data from the environment. The system dynamically determines the inter-dependencies between these decision variables and also periodically updates them into an aggregated decision outcome (prediction). This aggregated decision outcome is then evaluated with respect to the desired decision outcome, and, depending on the deviation between the actual and desired decision outcomes, actions are recommended to collect additional data/information and discover new data correlations. This information is then used to update the decision variables autonomously and proactively - so that the quality of the aggregated decision outcome successively converges towards the desired outcome. We plan to realize the aforementioned functionalities in CONRAD using three subsytems that are summarized below:

(1) Decision Making Subsystem: The decision making subsystem uses a prediction market-based information aggregation mechanism to update and mathematically combine or aggre-

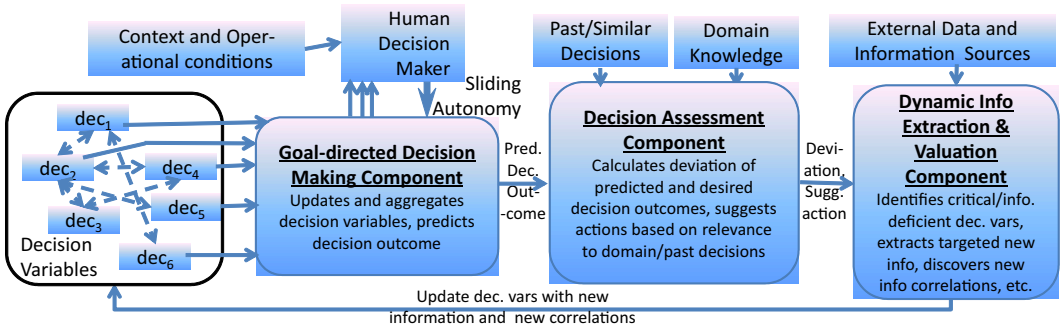


Fig. 1. Different components of the CONRAD system that integrate the decision making and network analytics aspects.

gate information from different decision variables and predict the outcome of the decision event.

(2) Decision Assessment Subsystem: The decision assessment subsystem uses the currently predicted decision outcome from the decision making subsystem along with relevant domain knowledge from past decisions made in similar domains to predict the decision trajectory and recommends information collection-related actions to the human decision maker. Machine learning and AI-based planning techniques are used to implement the functionalities of the decision assessment subsystem.

(3) Dynamic Information Extraction and Valuation Subsystem: The dynamic information extraction and valuation subsystem uses the actions recommended by the decision assessment subsystem to model and dynamically update the dependencies and correlations between events and people that influence the decision variables using metrics and techniques from large scale network analysis. The different components of CONRAD and their main functionalities are given in Figure 1 and discussed in the following sections.

II. Goal-Directed Decision Making

The main research question addressed in CONRAD's goal directed decision making subsystem is how to design a suitable set of computation techniques to dynamically update the different decision variables in the current decision context and combine or aggregate them into a single, global decision outcome. The decision variables are extracted from the environment's information by CONRAD's information extraction component, discussed in Section IV. We propose to perform the update and aggregation of the individual decision variables using an information aggregation technique inspired by prediction markets. Prediction market based information aggregation [14] has been recently shown to be a reasonably accurate means of predicting the outcome (usually binary or discrete valued outcome) of an event that is going to happen in the future. In our previous research, we have developed several successful techniques for multi-agent based prediction markets [8], [9] where the market's trading operations are performed by automated software agents. In prediction markets, information is collected from people, news sources, etc. in the form of bids, using either virtual or real money, on the possible outcome (binary-valued) of a future event. These bids are aggregated and the aggregated value represents the people's prediction of the event's outcome. A schematic of CONRAD's goal-directed decision making component is shown in Figure 2. To explain our approach, we use a few mathematical notions - let $\{dec_i^t\}$ denote the set of individual decision variables of the current decision making context at time step t, $AggDec^t$ denote the aggregated decision from aggregating $\{dec_i^t\}$ at time step t, where $dec_i^t, AggDec^t \in [0,1]$. With this formulation, each dec_i^t can be interpreted as a probabilistic confidence or belief of the decision variable; likewise for $AggDec^t$.

Dynamic decision variable updates. To enable dynamic updates of the decision variables, we associate each dec_i^t with a decision making (or belief update) agent a_i; a_i is responsible for updating the value of dec_i^t at time step t. Agent a_i performs this update using the following belief update formula:

$$dec_t^i = bel_i(dec_i^{t-1}, dec_{-i}^{t-1}, AggDec^{t-1}),$$

where $bel_i(.)$ is the belief update function used by agent i, dec_i^{t-1} is the value of dec_i during time step $t-1$, dec_{-i}^{t-1} is the set of decision variables from time step $t-1$, excluding dec_i^{t-1} itself, that are correlated with dec_i^t during time step t and $AggDec^{t-1}$ is the value of the aggregated decision outcome during time step $t-1$. The decision maker agent also ensures that decision variables that have already converged to their optimal or best value are not updated. The decision maker agent uses the intelligence, from reviewing the current context, to identify only those decision variables that need updating.

Aggregating decision variables. At the next step, the individual decision variables are combined into an aggregate or predicted decision outcome by the aggregator agent. A market-based aggregation mechanism provides a suitable way to combine information from multiple sources (e.g., multiple decision variables updated by the decision maker agents) into a single aggregated decision outcome value using a technique called a scoring rule [7].

III. Decision Assessment

The objective of CONRAD's decision assessment component is to determine how well the current aggregated (predicted) decision outcome is aligned with the desired decision outcome and to recommend actions related to future information collections that could potentially improve the convergence of the predicted decision outcome towards the decision outcome. A schematic of the decision assessment component is shown in Figure 3. Because we have represented decision outcomes as probability distributions (belief values), statistical divergence metrics such as the Kullback-Leibler (KL) divergence can be used to predict the future decision trajectory - Some

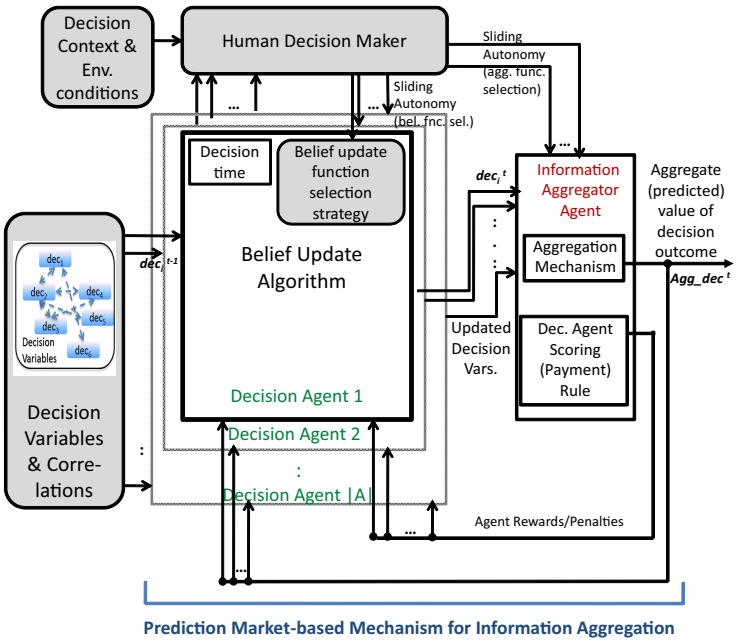


Fig. 2. The core of the prediction market based information aggregation technique used in the decision making subsystem of CONRAD.

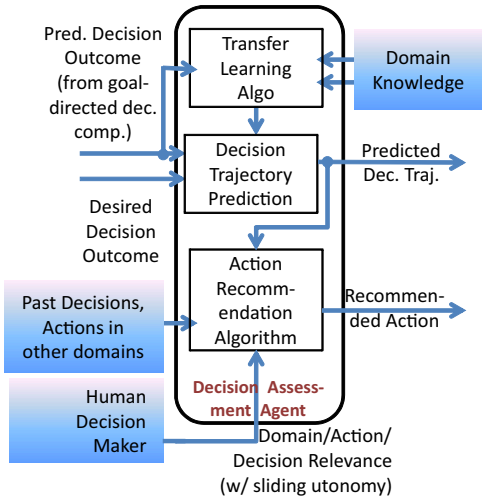


Fig. 3. Different components within the decision assessment subsystem of CONRAD.

well-known decision trajectories can be constructed from past decisions and then Bayesian inference can be used to classify the current decision trajectory into one of the trajectory types. The historical aggregated decisions can be further refined with domain knowledge to reflect the changes in the situation since the decisions were aggregated. The decision assessment subsystem also suggests actions related to future information collection to the human decision maker and to CONRAD's information extraction/evaluation component using AI-based planning techniques such as MDPs and POMDPs [12]. The outcome of the action recommendation algorithm would be a probabilistic distribution over recommended actions from which an action can be picked strategically by CONRAD's Information Extraction and Valuation component.

IV. Information Extraction and Valuation

The key to efficient decision making is to ensure that the available information is dynamically updated and important correlations in data are accurately captured. To achieve these objectives, CONRAD will perform the following operations in real time;

Extract Decision Variables from Raw Data. The data extraction tool of CONRAD extracts data from different heterogeneous and potentially changing sources and filters decision variables - id of the data creator, the data creation time, and a list of key fields such as demography, topic of discussion, etc. We will use Semantic Technology and represent the list of fields through an ontology based language such as OWL. Our goal is to create a database similar to DBpedia (dbpedia.org/About) that will allow users to submit queries with multiple conditions and identify entities that fulfill those queries. The correlations between the decision variables are modelled as networks (or graphs). The vertices in the network represent the entities and the edges represent the correlations. Using this collected data, the information component performs the following subtasks: *(a) Creating Multilevel Networks.* A network is created from the processed data as follows - one field in the dataset is identified as the entity variable and other selected field(s) as the relation variable(s). Each vertex of the network represents a unique instance of the entity variable (here each entry is the name) and two entities are connected if they satisfy certain relations between the relation variables (for example, ids with age difference of five years or less are connected). The connectivity patterns of the networks can with the time stamp changes. Networks based on the same entity variable can be further combined to a multilevel network. This enables us to unearth obscure information that is not immediately relevant from only one database. *(b) Real Time Analysis of Networks.* The analysis of the networks provides insights to the characteristics of the data. Some of the common analysis objectives in CONRAD include (i) detecting communities to identify tightly connected groups of vertices [10] and (ii) computing centrality metrics, core numbers and driver nodes to determine the influential people (or data) [11]. We plan to extend these analysis by including the semantics of the networks. The edges in the network will be annotated by their semantic values (i.e. age, demography, etc). We can therefore refine communities obtained from the initial vertex based method combining entities that have similar semantic values in their links.

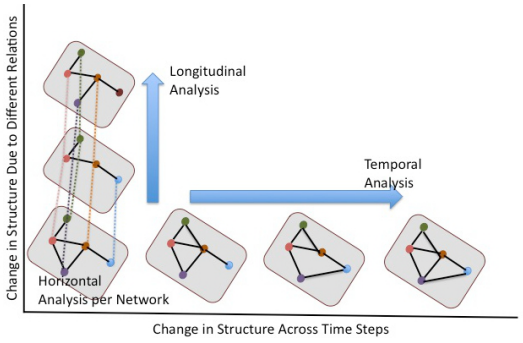


Fig. 4. Analysis of network models over three levels; vertices of the same color represent the same entity

CONRAD performs network analysis operations at three levels, as illustrated in Figure 4. The first is the horizontal level that analyzes each entity network. The second is the longitudinal level where the analysis is conducted across levels (the networks at each level have the same entity variable, but the relational variable(s) and therefore the structure is different). The third level is the temporal level where we track the changes to the network structure across different time steps [2]. CONRAD will implement parallel algorithms and approximate methods to perform the analysis in real time [1], [13]). The information network is connected to the decision variables by matching the component networks, each representing a decision variable to the appropriate decision making agent. For example, if the agent's decision is to deliver supplies to disaster stricken areas, then the agent has to obtain information from networks whose entity variable is the location as well as from the network whose entity variable is the demography.

Identifying Critical Decision Variables. Identify important decision variables that can predict future events will enable users to maintain the correct decision trajectory. The critical variables are the ones whose corresponding networks guarantee that the analysis results are accurate under various perturbations to the data and are sensitive to changes in the data. CONRAD evaluates the reliability of the network models based on *well-posedness* and the *sensitivity* with respect to the analysis objective. To the best of our knowledge, our work is one of the first instances that a network analysis toolkit will include a component to compute the accuracy of the data. *Well-posedness* is a measure of whether the analysis objective, is feasible for a given network. To compute well-posedness of a network, CONRAD computes the number of solutions that the network has for a given analysis function. For community detection, this can be computed by changing the vertex ordering, and then taking the consensus of the communities obtained at each ordering to find the well-posed subgraph [4]. This computation can be extended to the overlapping communities as well. For centrality metrics and core numbers, we are interested in only the high valued ones. To determine whether a network is well-posed, the centrality values for each vertex is first evaluated and then the size of the set of 'high-valued' vertices is checked. If this set is very large, then none of the vertices will be distinctively important. *Sensitivity* measures whether a small change to the input produces a commensurate change in the results. To compute the sensitivity of network analysis, CONRAD uses models of small perturbations (or noise) to the network and metrics to evaluate this noise [6]. After evaluating all the networks based on these well-posedness and sensitivity, the system will retain only the ones that produce accurate results and are sensitive to changes in data. The entity and relation variables of these networks are the critical variables and will produce reliable data patterns that can be used for prediction.

Integrating Decision Making and Data Extraction. The final objective of the information extraction and valuation component in CONRAD is to use the recommendations from the decision assessment algorithm to update the information networks. Based on the recommended actions, the information component tries to extract 'meaningful information' from 'raw data'. The main operations of this process are (i) improving the data gathering mechanism, (ii) improving the quality of the networks and (iii) improving accuracy of the analysis.

Data Gathering. The data gathering operation can be improved by adding more varied sources of information. For example, we can enrich information about possible disasters, by including information of past hurricanes and earthquakes, in addition to tracking the current disaster through news sources, and social network sites.

Adaptive Refining of Data Data is generally gathered 'wholesale', without specifically considering the subsequent use of the information. In the network modeling stage, the system refines the data by filtering the initial network based on certain combinatorial properties. For example if the agents' focus is on finding clusters of similar entities, then a chordal graph based filtering that will retain only the tightly connected components in the network is used. Conversely if the decisions are to be based on centrality metrics, then filtering to reduce the low weight edges is more effective [5].

V. Conclusion

In this position paper, we have proposed CONRAD, a real-time, proactive decision aiding tool that leverages the advantages of game theory, machine learning and network analysis. Each of the individual components proposed for CONRAD have been shown to be successful in their respective domains and we posit that combining them will further enhance the decision making capabilities.

References

[1] D. Bader, S. Kintali, K. Madduri and M. Mihail. Approximating Betweenness Centrality. *The 5th Workshop on Algorithms and Models for the Web-Graph* (WAW2007), 4863, 134, 2007.

[2] S. Bansal, S. Bhowmick and P. Paymal. Fast Community Detection For Dynamic Complex Networks. *Communications in Computer and Information Science* Vol. 116, 196-207, 2010.

[3] F. Burstein and C. Holsapple (editors), "Handbook on Decision Support Systems 1: Basic Themes,", International Handbooks on Information Systems, Springer, 2008.

[4] T. Chakrabory, S. Srinivasan, N. Ganguly, S. Bhowmick, A. Mukherjee. Constant Communities in Complex Networks. *Nature Scientific Reports* 3, Article number: 1825 doi:10.1038/srep01825 2013.

[5] K. Dempsey, S. Bhowmick and H. Ali. Function-preserving Filters for Sampling in Biological Networks. *Procedia Computer Science*,Vol. 9, 587-595, 2012.

[6] A. Flaxman and A. M. Frieze. The diameter of randomly perturbed digraphs and some applications. APPROX-RANDOM, 2004.

[7] R. Hanson, Logarithmic Market scoring rules for Modular Combinatorial Information Aggregation. *Journal of Prediction Markets*, 1(1):3-15, 2007.

[8] J. Jumadinova, and P. Dasgupta. Distributed Decision Making in a Multi-agent Prediction Market using a Partially Observable Stochastic Game. *Proceedings of the 13th International Conference on E-Commerce (ICEC). Liverpool, UK,*:21:1-21:10, 2011

[9] J. Jumadinova and P. Dasgupta, A Multi-Agent System for Analyzing the Effect of Information on Prediction Markets. *Intl. J. of Intell. Systems* 26(5):383-409, 2011.

[10] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E.* 69, 2004.

[11] Y.-Y. Liu and J.-J. Slotine and A.-L. Barabasi. Controllability of complex networks. *Nature* 473, 167173. 2011. doi:10.1038/nature10011

[12] S. Russell and P. Norvig. Artificial Intelligence: A Modern Approach, Prentice Hall, 2009.

[13] V. Ufimtsev and S. Bhowmick, Application of Group Testing in Identifying High Betweenness Centrality Vertices in Complex Networks. *11th Workshop on Machine Learning with Graphs, KDD* 2013.

[14] J. Wolfers and E. Zitzewitz. Prediction Markets. *Journal of Economic Perspectives*, 18(2):107-126, 2004.

Need for Community of Interest for Context in Applied Decision Making

Warfighter's Use of Context for Decision Making

Peter S. Morosoff
President, Electronic Mapping Systems, Inc. (E-MAPS)
Fairfax, VA, USA
Peter.Morosoff@e-mapsys.com

Abstract - **There is interest in building a community of interest for Context in Applied Decision Making. Warfighters have long exploited context in decision making. The mystery, therefore, is why the information technology (IT) community that supports warfighters provides so little IT that exploits context for decision making. One possible answer is the lack of a forum such as a community of interest that facilitates sharing (a) among those who do or might develop IT that exploits context for decision making and (b) with warfighters. This paper provides background information on warfighter's use of context and highlights an IT system that uses computer representations of context in order to facilitate establishing a community for Context in Applied Decision Making.**

Keywords – **ontology; context in decision making; warfighters; ICODES**

I. BACKGROUND

A community is needed for Context in Applied Decision Making because warfighters rely on context when processing data to create information and make decisions required for mission accomplishment. Further, for the last 20 years, warfighters and IT specialists have collaborated to create and evolve (a) at least one program of record (POR) IT system that processes data into information based on context and (b) several such applications for advanced concept technology demonstrations (ACTD) and other science and technology (S&T) efforts. Documents such as the 1998 presentation "Coping with Massive Amounts of Information: The Glare of War" produced and shared by Dr. Howard Marsh of the Office of Naval Research (ONR) are now impossible to locate. For the last 15 years, we should have been building on Dr. Marsh's insights. Instead, we continue to invest effort in replicating his research.

DoD needs the subject community of interest so DoD can shift from fragmented, individual successes that are rarely exploited in later efforts to an effective system in which (a) new successes build on earlier successes, (b) new successes avoid the problems of past failures, and (c) warfighters, who need IT tools that use context in processing data to produce information needed for good decisions, can readily share their needs, circumstances, and constraints with developers.

Because of reductions in DoD funding, there is a pressing need to not repeat mistakes made in earlier IT programs and to provide useful products as rapidly as possible. Indeed, simply making information on existing POR tools that exploit context in decision making easily available may be the most important short-term product of this community.

II. WARFIGHTERS' USE OF CONTEXT

People in general seem to be naturally inclined to focus their own contributions to current problems and to be unaware of and give proper credit to the intellectual and organizational accomplishments of past commanders and others. The more data and information that is generated and available, the harder it is to find relevant information. Napoleon is an example of an individual who was remarkably successful at creating a mobile capability to (a) assemble and move with him maps, files, and other information that provided him with context that he could (b) then use in processing incoming reports and other data to create the information he needed for battlefield successes. However, Napoleon's accomplishments in this area are also largely unknown. Anders Engberg-Pedersen writes in his dissertation "*The Empire of Chance. War, Literature, and the Epistemic Order of Modernity*" [1] that:

> Two wagons served the transportation of these maps, and later a lighter cabriolet was added due to its greater speed. Moreover, Napoleons own wagon was converted into a rolling office: drawers were installed for a small reference library where he would also store reports from Paris. When the drawers were full, superfluous material was cut into pieces and thrown out the window, which, according to Odeleben, could result in a veritable "paper rain." [2]. A central concern was thus to organize the cartographic material in a practical way in order to make it transportable and readily available.

Infantry in Battle [3], a book produced under the direction of George C. Marshall when he was a colonel leading the

Army's infantry school, is very clear on the value of understanding context when considering data and information. Chapter V, "Terrain," opens with the statement "Maneuvers that are possible and dispositions that are essential are indelibly written on the ground." That is, the terrain is a context for ground operations that, if understood, facilitates (a) predicting what enemy can and might do and (b) what our forces would benefit from doing and must do.

My favorite example of a warfighter using context is when US Marine Corps Captain Frank Izenour determined the start date of the major 1972 North Vietnamese offensive - now known as the Easter Offensive. In the course of working with Capt. Izenour from 1982-6, I learned the specifics from him directly. That he, in fact, made the prediction before the attack is documented in Marine Corps Colonel Gerald Turley's book, *The Easter Offensive* [4]. Early in that book, while Turley is recounting his early days with the Marine Advisory Unit in Vietnam, he states that Capt. Izenour was convinced the North Vietnamese would attack sooner rather than later.

How did Capt Izenour use context to predict what so many more experienced and senior officers missed? The most important element, as I learned from working with him, was that Capt. Izenour was a reader and a thoughtful officer. When he got data and information, he thought about them and searched for implications and logical conclusions. In early 1972, his assignment provided him access to a U.S. intelligence center in Saigon where he viewed large maps that used icons to represent the locations of North Vietnamese Army (NVA) units across and outside South Vietnam. These maps showed NVA units positioned the length of South Vietnam's boarders with its neighbors. The locations of these units, along with the resources required to deploy and support them in the field, produced information context that suggested to Capt. Izenour that the NVA was planning an attack across all of South Vietnam. The question was when, not if, a major country-wide attack would be launched.

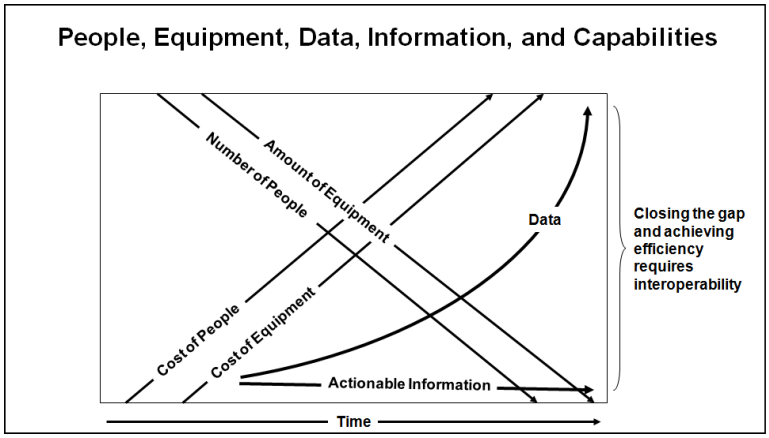
Capt. Izenour told me that opinions as to when the attack would come were varied. August and September 1972 were favored by many people with access to the intelligence. However, Capt. Izenour's information context included the monsoon seasons in South Vietnam. The monsoon comes to southern and northern South Vietnam at different times. The only period the southern and northern parts of the country were not having monsoons was in the three months of March through the end of May. Given that context, Capt. Izenour calculated the NVA would allow 30 days for the ground to dry and then launch an attack about April 1, 1972 across all of South Vietnam. In the actual event, he was off by only 24 hours. Unfortunately, because so few others shared his context and opinion, the Easter Offensive was a strategic surprise for the U.S. and significantly advanced the NVA's objective of gaining control of South Vietnam.

III. Operation Desert Shield and Storm: Data Overwhelms Context

In early 1991, the author of this paper was sent to Saudi Arabia to conduct a Marine Corps battlefield assessment of command and control in Operation Desert Storm. The author arrived shortly after the fighting ended and started interviewing participants in the war. To the author's surprise, those he interviewed who had served in the Vietnam War kept noting that fewer people and less equipment had been provided for tasks in Operation Desert Storm than the same tasks in the Vietnam War. Dr. Katherine McGrady, of the Center for Naval Analyses (CNA), had been detailed to support I Marine Expeditionary Force (I MEF) during Operation Desert Storm. When asked about the less equipment and fewer people in Desert Storm than in Vietnam, Dr. McGrady replied that the salaries of people and the cost of equipment were rising while manpower was being reduced and the new equipment being fielded was more capable than the equipment it replaced. The ongoing result was that senior leaders were counting on the fewer people being able to make better decisions so that better operational effects could be created with fewer pieces of better equipment.

Additionally, the war participants discussed the volume of data forced upon them. The G-2 (i.e., intelligence officer) stated that on the busiest days of the fighting, the intelligence section received so many reports that they stopped counting them at 6,000 a day, and they could not and did not even read all the 6000+ messages on those days.

This led the author to develop the following drawing depicting rising salaries and increasing cost of equipment with decreasing numbers of people and pieces of equipment as data volume increases at an ever-faster rate. The conclusion is that future IT after Operation Desert Storm would need the capability to process ever-increasing volumes of data into less but better focused information that commanders would need to make better decisions and produce better results with fewer pieces of equipment. If better IT was not produced, the cost of the people needed to process the available data would make DoD unaffordable.



IV. USE OF IT TO EXPLOIT CONTEXT

We now turn to successes in developing IT that exploits context warfighters use.

During Operations Desert Shield and Storm, U.S. forces deployed to Saudi Arabia by ship. The process and methods for planning ship loads was well developed by the start of Operation Desert Shield. Stripped to its essentials, planning a ship load is an exercise in determining where to place equipment of known dimensions using the context provided by a ship's plan (e.g., dimensions of a ship's storage areas and ramps). Given sufficient time, skilled load planners could develop good load plans manually.

However, Operations Desert Shield and Storm revealed that no-notice wars such as the Gulf War provide insufficient time for manually planning and adjusting ship load plans as the situation develops. The fog of war extended to the deployment of forces. Units found that the transport ships they had been told would carry their equipment and for which they had prepared load plans manually were replaced by other ships with little or no notice. The context or layout of the new ship could be learned easily, but often there was insufficient time to prepare a good load plan manually for the replacement ship.

After Operation Desert Storm, the Army's Military Traffic Management Command (MTMC), the command responsible for loading military equipment on ships, sought to develop IT support for agile load planning for ships. The objective was to extend the context from people-based activities to computer-based activities. These agile load planning inquiries were answered by the Collaborative Agent Design Research Center (CADRC) at the California Polytechnic State University (Cal Poly) at San Luis Obispo, California. For several years, Dr. Jens Pohl and his associates in CADRC had been experimenting with using ontologies to represent context and collaborative software agents to exploit the context provided by ontologies. When data on the equipment to be loaded on a ship was entered into the IT application, software agents would process the data based on the ontology(ies) and quickly develop an effective load plan [5].

The early experiments for MTMC matured into an application that was first fielded in 1997 under the name Integrated Computerized Deployment System (ICODES). In the intervening quarter century, ICODES has continually evolved with its latest version operating in a cloud environment.

ICODES' use of ontology and software agents has also been exploited in the Extending the Littoral Battlespace (ELB) Advanced Concept Technology Demonstration (ACTD), the Coalition Secure Management and Operations System (COSMOS) ACTD, and other efforts.

V. CONCLUSION

The role of context in applied decision making is well established and there is a rich body of literature on the subject. ICODES has demonstrated the efficiencies and increased effectiveness possible when context is exploited in IT systems used by warfighters. A forum such as a COI is needed that facilitates IT developers and others accessing literature and each other. From the perspective of a community on Context in Applied Decision Making, ICODES is important because its results include (a) significant reductions in the time to plan a ship load, (b) improved detection of potential hazardous materials violations, (c) significantly fewer senior ship load planners, (d) reductions in rental expenditures for piers and staging areas for loading military equipment onto ships and (e) effective use of applied ontologies and software agents. From the perspective of DoD, a community of interest is important because it would facilitate the exploitation of past successes and collaboration among ongoing and future efforts while contributing to better DoD efficiency.

REFERENCES

[1] Anders Engberg-Pedersen, "The Empire of Chance: War, Literature, and the Epistemic Order of Modernity." Dissertation, Harvard University, 2012.

[2] Otto von Odeleben, "Napoleons Feldzug (Expedition), Abschnitt (Section) 153".

[3] Infantry in Battle, 1934, War Department, Washington, DC.

[4] Gerald Turley, "The Easter Offensive: The Last American Advisors Vietnam, 1972". Presidio Press, Novato, CA

[5] Kym Pohl and Peter Morosoff, "ICODES: A Load-Planning System that Demonstrates the Value of Ontologies in the Realm of Logistical Command and Control (C2)," InterSymp-2011, Baden-Baden, Germany, August 2, 2011.