SEMANTIC TECHNOLOGY FOR
INTELLIGENCE, DEFENSE, AND SECURITY

STIDS 2014

Semantics in Support of Collaboration

THE 9TH INTERNATIONAL CONFERENCE
ON SEMANTIC TECHNOLOGY
FOR INTELLIGENCE, DEFENSE AND SECURITY
NOVEMBER 18-21, 2014
Johnson Center
George Mason University
Fairfax, Virginia Campus

Conference Proceedings

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

GEORGE MASON UNIVERSITY

C4I CENTER

CRAY

Preface

The 9th 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 2014 theme is Semantics in Support of Collaboration.

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 2014

Ian Emmons and Kathryn Laskey
STIDS 2014 Technical Chairs

Paulo Costa
STIDS 2014 General Chair

STIDS 2014 Committees

STIDS 2014 Program Committee

Carl Andersen	Raytheon BBN Technologies
Rommel Carvalho	Brazilian Office of the Comptroller General and University of Brasilia
Werner Ceusters	SUNY at Buffalo
Paulo Costa	George Mason University
Timothy Darr	Knowledge Based Systems Inc.
Mike Dean	Raytheon BBN Technologies
Ian Emmons	Raytheon BBN Technologies
Matthew Fisher	Progeny Systems
Katherine Goodier	Xcelerate Solutions
Mark Greaves	Pacific Northwest National Laboratory
Richard Haberlin	EMSolutions Inc.
Peter Haddawy	Mahidol University
Brian Haugh	IDA
John Hebeler	University of Maryland, Baltimore County
Terry Janssen	Quantum Cybersecurity Systems LLC
Gregory Joiner	Raytheon BBN Technologies
Anne-Laure Jousselme	NATO Centre for Maritime Research and Experimentation
Mieczyslaw Kokar	Northeastern University
Dave Kolas	Raytheon BBN Technologies
Kathryn Laskey	George Mason University
Nancy Lawler	U.S. Department of Defense, retired
Louise Leenen	CSIR
Michael Letsky	MacroCog Associates
William Mandrick	Data Tactics
Daniel Maxwell	KaDSci, LLC
David Mireles	Raytheon BBN Technologies
Ranjeev Mittu	US Naval Research Laboratory
Jeffrey Morrison	Office of Naval Research
Leo Obrst	MITRE
Alessandro Oltramari	Carnegie Mellon University
Mary Parmelee	MITRE
Gregor Pavlin	Thales Group
Andrew Perez-Lopez	Opower
Plamen Petrov	Raytheon BBN Technologies
Setareh Rafatirad	George Mason University
Douglas Reid	Google
Dorene Ryder	Raytheon BBN Technologies
Robert C. Schrag	Haystax Technology
Patrice Seyed	3M HIS
Ciara Sibley	Naval Research Lab
Barry Smith	National Center for Ontological Research, University at Buffalo

Kathleen Stewart	University of Iowa
Gheorghe Tecuci	George Mason University
Brian Ulicny	Thomson Reuters
Amanda Vizedom	Criticollab, LLC
Andrea Westerinen	Nine Points Solutions, LLC
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	Xcelerate Solutions
Terry Janssen	Quantum Cybersecurity Systems LLC
Kathryn Laskey	George Mason University
William Mandrick	Data Tactics
Leo Obrst	MITRE Corporation
Barry Smith	NCOR, University at Buffalo

STIDS 2014 Organizing Committee

General Chair
Paulo Costa

Technical Chairs
Ian Emmons
Kathryn Laskey

Publicity Chairs
William Mandrick
Alessandro Oltramari

Classified Session Chair
Brian Haugh

Local Team (GMU)
Debra Schenaker (Administrative Chair)
Priscilla McAndrews
Shou Matsumoto
Felipe Bombarda Guedes
Tamara Day

STIDS 2014 Platinum Sponsor

Global supercomputing leader Cray Inc. (NASDAQ: CRAY) provides innovative systems and solutions enabling scientists and engineers in industry, academia and government to meet existing and future simulation and analytics challenges. Leveraging more than 40 years of experience in developing and servicing the world's most advanced supercomputers, Cray offers a comprehensive portfolio of supercomputers and big data storage and analytics solutions delivering unrivaled performance, efficiency and scalability. Cray's Adaptive Supercomputing vision is focused on delivering innovative next-generation products that integrate diverse processing technologies into a unified architecture, allowing customers to meet the market's continued demand for realized performance. Go to www.cray.com for more information.

Michael Dean Best Paper Award

August 7, 1961 - November 19, 2014

The Michael Dean Best Paper Award was established in 2014 in recognition of Michael Dean's many and diverse contributions to the STIDS community. We are gratified that Mike was aware that an annual award was being established in his honor, and deeply saddened at his passing one day prior to announcement of the winner. This year, at this sad moment, the award carries special meaning, both for the winner and for the entire STIDS community.

In selecting the winner, the committee sought to highlight the qualities that made Mike such an asset to the community. The criteria for selection exemplify the very best contributions to the conference and the community.

To this end, the Michael Dean Best paper is the paper that, in the judgment of the award committee, best satisfies the following criteria.

1. Conveys a clear, careful understanding of the problem or issue being addressed, and clearly states why it matters.

2. Conveys a thorough understanding of technical issues, and a well-grounded, pragmatic view of prior and related work.

3. Clearly identifies the specific semantic technologies being discussed, and their relationship to the problem.

4. Identifies specific experience or expertise on which the paper and its conclusions draw.

5. If a semantic system or application is being presented as part of a solution, clearly identifies and communicates the components of this system, including any ontologies, and how they interact, as well as their degree of actuality, availability, maturity and source.

6. Identifies whether and how such system/ application/ components have been evaluated and with what results.

7. Identifies outcomes, experiences, and lessons learned.

8. Demonstrates prioritization of greater technical and domain understanding and problem-solving over self-promotion, organization promotion, partisan or programmatic scorekeeping, or other, narrower concerns.

9. Demonstrates knowledge of prior and current art, strengthens such knowledge in the community, and promotes better understanding by sharing the rationale for choices, especially when they diverge from common practice.

10. Demonstrates and strengthens the state of the art of Semantic Technologies via the quality of the work described. Provides promising ways forward while negotiating known trade-offs and avoiding known pitfalls. Helps more junior technologists avoid repetition of old errors, and provides more senior technologists with new insights.

The winner of the award was announced at a special session on the last day of the conference.

- *2014 Michael Dean Best Paper*: Daniel L. Costa, Matthew L. Collins, Samuel J. Perl, Michael J. Albrethsen, George J. Silowash, Derrick L. Spooner. 2014. An Ontology for Insider Threat Indicators.
- *First runner-up*: Alessandro Oltramari, Lorrie Faith Cranor, Robert J. Walls, Patrick McDaniel. 2014. Building an Ontology of Cyber Security.
- *Second runner-up*: Robert C. Schrag, Edward J. Wright, Robert S. Kerr, Bryan S. Ware. 2014. Processing Events in Probabilistic Risk Assessment.

2014 Michael Dean Award Committee

Leo Obrst (Chair)	MITRE Corporation
James Hendler	Rensselaer Polytechnic Institute
Ian Horrocks	University of Oxford
Dave Kolas	Raytheon BBN Technologies
Amanda Vizedom	Criticollab, LLC

STIDS 2014 Keynote Speaker: November 19

Dr. Mark Greaves

Accelerating Discovery in the 21st Century

One of the most technically promising areas for semantic technology has been in accelerating the process of scientific discovery. Like the STIDS community, the general scientific community has struggled with needing to combine and understand data in support of modeling and data analytics. Semantic technology is frequently part of the answer. However, advanced scientific information processing systems have gone beyond supporting data integration and modeling, to attempt to directly assist scientists in creating and testing hypotheses. In this talk, I will discuss some new developments and challenges in AI systems that support scientific discovery.

I will also discuss several parallels to the concerns of the STIDS community, and conclude with some lessons.

Biography: Dr. Mark Greaves

Mark Greaves is currently Technical Director for Analytics in the National Security Directorate of the US Department of Energy's Pacific Northwest National Laboratory, providing scientific, programmatic, and business development leadership. At PNNL, he works with national security clients and cutting-edge PNNL scientific teams to advance the nation's overall capability for extracting meaning from large heterogeneous data sets.

Previously, Mark was Director of Knowledge Systems at Vulcan Inc., the private asset management company for Paul Allen, where he led advanced research teams in large knowledge bases and semantic web technologies, including semantic wikis and data-intensive artificial intelligence technologies. He also served as Vice President of the Allen Institute for Artificial Intelligence, which supports individual researchers and research groups that have the promise to dramatically accelerate progress in artificial intelligence. Prior to Vulcan, Mark was Director of DARPA's Joint Logistics Technology Office, and Program Manager in DARPA's Information Exploitation Office. At DARPA, he directed national research programs in semantic web technology, formal ontology specification, logistics and supply chain control technologies, and the application of software agent technology to problems of distributed control of complex systems-of-systems. In May of 2005, he was awarded the Office of the Secretary of Defense Medal for Exceptional Public Service for his contributions to US national security while serving at DARPA.

Prior to coming to DARPA, Mark worked on advanced programs in software agent technology at the Mathematics and Computing Technology group of Boeing. He has published two books and over 20 papers, holds two patents, chaired t he FIPA technical committee on agent communications languages, and from 2001 to 2004 served as co-chair of the Joint Readiness and Sustainment of Strategic Systems chapter of the Defense Joint Warfighting Science and Technology Plan. He currently serves on the Advisory Boards of several Semantic Web-oriented organizations, and is a Fellow of the Semantic Technology Institute. Mark holds a PhD in Philosophy from Stanford University.

STIDS 2014 Keynote Speaker: November 20

Dr. Heather McCallum-Bayliss

Meaning Technologies: At the Cutting Edge

Exploring new types of knowledge that can be extracted from on-line data is crucially important to US intelligence, defense and security concerns. In general, we continue to move existing technologies into different corners and aspects of the data world but we have not explored very successfully the development of technologies that tackle the areas of cognition and social interaction. Inferring behaviors and intentions is currently dependent on external criteria or ontologies but there is often little motivation or evidence. Two IARPA programs have ventured into uncharted territories by attempting to determine if on-line data have more to reveal than we have considered. The SCIL Program is based on the hypothesis that automated systems can identify the strategies that people interacting in on-line groups use to establish their social roles and signal their social intentions by examining the language they use. The Metaphor Program is based on the hypothesis that automated systems can reveal the underlying beliefs and worldviews of cultures by examining their use of metaphorical language. The talk will delve into the status and results of these research efforts. Information about IARPA, the goals of the organization and principles that guide research programs will also be introduced.

Biography: Dr. Heather McCallum-Bayliss

Heather McCallum-Bayliss received her Ph.D. in Theoretical Linguistics from Georgetown University. She taught at Georgetown for seven years before moving to a research firm that focused on the development of human language technology systems with particular focus on systems for multilingual and multicultural name-searching. This work resulted in a patent for a phonologically based, multicultural name-search system. Since 2003, Dr. McCallum-Bayliss has defined and managed advanced multilingual research programs at the ODNI (Office of the Director of National Intelligence)'s Intelligence Advanced Research Projects Activity (IARPA) and its predecessor organizations. These efforts have produced automated question-answering systems as well as systems that tackle the use of language for characterization of social and cultural meaning and understanding. Dr. McCallum-Bayliss continues at IARPA as the Program Manager for the Metaphor and SCIL Programs.

STIDS 2013 Invited Speaker: November 20

Dr. Jeffrey Morrison

Uncertainty in Decision Making

The science & technology community faces a challenge – and an opportunity. Much research has addressed helping decision makers "Know what they Know". We now need to think about helping decision makers in making decisions with incomplete, or imperfect information, i.e. making decisions in uncertainty. The semantic technology should play a key role in There is an urgent need to find ways to address this challenge. Many DOD mission areas need to Maximize the Impact of Limited Operational Assets allocated within a dynamic and uncertain "targeting" environment. This presentation will propose a number of questions, including:

- How might we develop strategies for Managing Uncertainty?
- Can we Optimize given the uncertainties?
- Can we Develop Quantitative Decision Support to Manage Uncertainty?

Research is needed that takes a different direction from much of the work with uncertainty to date. Rather than developing estimates of uncertainty, what is needed are decision support tools to optimize decision making despite uncertainties. Insights will be provided on the role of Uncertainty in the "targeting" process for targeteers, mission planners, and watchstanders to determine when & how to task operational assets. The Office of Naval Research has begun several projects to understand operational requirements for making decisions in uncertainty for Navy operations, and representative decision support concepts, and modelling efforts currently being developed are described.

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 a 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

Technical Papers

Position Papers

Technical Papers

An Ontological Approach to Territorial Disputes

Brian Donohue, J. Neil Otte, and Barry Smith
Department of Philosophy, University at Buffalo,
Buffalo, NY, USA
{bd26, jeffotte, phismith}@buffalo.edu

Abstract – **Disputes over territory are a major contributing factor to the disruption of international relations. We believe that a cumulative, integrated, and continuously updated resource providing information about such disputes in an easily accessible form would be of benefit to intelligence analysts, military strategists, political scientists, and also to historians and others concerned with international disputes. We propose an ontology-based strategy for creating such a resource. The resource will contain information about territorial disputes, arguments for and against claims pertaining to sovereignty, proffered evidence for such claims, political and military motives (overt or hidden), and associated conflicts. Our approach is designed to address several issues surrounding the representation of geopolitical conflict, including the tracking and individuation of disputes and the validation of disseminated information.**

Keywords—applied ontology; territory; international conflict; BFO 2.0

I. Introduction

In what follows, our focus is on the ontological representation of territorial disputes and the different sorts of entities associated therewith. The ontology is intended to be used for the consistent annotation of data and information about territorial claims, arguments for and against such claims, political and military motives (overt or hidden), and the different types of conflicts associated with territorial disputes, from occasional skirmishes and limited engagements to terrorist campaigns and outright war.

We begin with a survey of some of the problems faced in representing territorial disputes in data and information systems. Second, we sketch how Basic Formal Ontology (BFO) 2.0 can be used as a framework for a realist understanding of different ways in which individual and group agents participate in territorial and other conflicts (http://www.ifomis.uni-saarland.de/bfo/). Third, we provide a framework for the handling of data about territorial claims; such claims can cause problems for a realist ontology, since they are often marked by the use of empty or disputed reference, for example when opposing parties in a territorial conflict produce maps of putative political entities in a given territory which cannot simultaneously be veridical. Finally, we propose a way to capture the relationships between motives and arguments underlying territorial claims, and we conclude with a case study of a territorial dispute between Japan and Russia.

II. The Trouble with Territorial Disputes

The problems facing the successful ontological representation of territorial disputes include:

(1) The need to identify the dispute at issue (which may evolve with time and may be described in conflicting ways by the disputing parties, as when Israelis use ‘Israel’ and Palestinians use ‘the Zionist entity’ to refer to the same territory).

(2) The need to identify the entities involved in a dispute, including:

- The disputing parties (governments, underground militias, liberation movements); leaders and representatives of the disputing parties (politicians, terrorist leaders, tribal leaders, religious leaders…).
- External or third parties (multinational agencies such as the UN, NGOs, aid agencies; mercenaries, religious bodies, press and broadcasting agencies, salient minorities (for example, Russians in East Ukraine).
- Actions (negotiations, propaganda, military actions, acts of terrorism, terrorist campaigns).
- Territories and territorial borders.
- Information artifacts involved in territorial disputes such as diplomatic notes, treaties, claims, maps.
- Objectives and motives of the parties involved (both overt and concealed).

III. Identifying and Tracking Disputes

The first step toward the representation of a territorial dispute is the ability to identify the dispute itself. This can be problematic, since how we should individuate any given dispute may be unclear. For example, is the on-going dispute over oil resources in the Arctic a *single* dispute or an interconnected web of disputes, involving multiple different countries? (See Supplementary Material Figure 4)

A second task for an ontology of territorial disputes is the need to take account of conflicting descriptions of both the dispute and the surrounding matters of (actual and purported) fact. For instance, the People’s Republic of China claims the island of Taiwan as one of its provinces; the inhabitants of Taiwan, on the other hand, maintain that, not only the island, but also the whole of mainland China, fall under the jurisdiction of a distinct sovereign nation, *viz.*, the Republic of

China, which was (they allege) wrongly forced off the mainland in 1949. Such incompatibilities cause problems for theories – like the theory defended by John Searle – which see political entities such as nations as social objects maintained in existence by the beliefs of the parties involved. An approach like Searle's, it seems, can provide an account of the ontology of geopolitical entities that is satisfied only in a world in which territorial disputes do not exist [1].

Because territorial disputes do indeed exist, as also do the associated conflicting claims, an ontology of territorial disputes must find some way to do justice to the fact that given claims may have no referent in the real world. To capture, for example, what is involved when disputants talk about "the Chinese territory of Taiwan" or "the Taiwanese territory of China," or when a neutral observer talks about the disputed territory of Taiwan combined with (the rest of) China, we need to find a way to link claims to the corresponding geographical regions without also prejudicing our representation in favor of one or other party.

Something similar holds when claims issued in the course of a dispute come in the form of accounts of a nation's historical presence in a territory which are offered as evidence to legitimize a claim of present sovereignty over that region (for example, accounts of the Jewish presence in the territory of present-day Israel since the time of the First Temple). Again, our ontology would need to be able to represent the content of such reports without necessarily endorsing their claim to truth, and the same would hold of geographic or oceanographic reports documenting measurements of the boundaries of given territories in ways that may affect claimsof sovereignty. For example, Russia has claimed approximately half of the Arctic Ocean (1.2 million square kilometers) on the basis of measurements of its continental shelf (the natural prolongation of its landmass, as defined by the United Nations Convention of the Law of the Sea (UNCLOS) – see [2]). The international community has greeted these measurements with considerable skepticism. Nonetheless, it is essential that analysts be able to search for data about the region Russia *does* claim. How, then, do we make sense of the content of our words without thereby imparting to our words a referent in the world? In what follows, we propose a solution to this problem, whose goal is to render an ontology capable of handling both true and false claims made within a territorial dispute.

IV. Entities in a Territorial Dispute

One of the questions an ontology is designed to answer is: What kind of entity is *X*? Consider for example the *role* of being an arbitrator of a dispute. In BFO terms, this is a specifically dependent entity – it could not exist apart from the person who bears the role. Or consider the information about some arbitrator contained in some document; to what kind of entity are we referring when we refer to *this information*? From the BFO point of view, an information artifact of this sort is a generically dependent entity, which means that whereas it requires some bearer – for instance, some hard drive – in order to exist, it does not require any specific bearer, because it can be copied from one hard drive to another; forwarded over e-mail, or printed out in the form of a paper document (http://code.google.com/p/information-artifact-ontology/).

In addition, an information artifact is a kind of entity that is capable of being *about* other entities. Territorial claims, maps, treaties—all exist as information artifacts in this sense. They are, like persons and roles, territories and territorial boundaries, what BFO calls continuant entities, which means that they continue to exist through time, even while undergoing changes of various sorts, for example in the form of amendments, codicils, and ratifications.

Territorial *disputes*, in contrast, are occurrent entities, which means that they are entities that occur in time and unfold themselves in successive temporal parts. Territorial disputes will differ along a number of dimensions, including their duration, the parties involved, and the degree to which they involve different levels of violent conflict.

They will also differ according to the territories to which they relate, the boundaries of these territories, the populations of human beings occupying these territories, and so on. An ontology to support reasoning with territorial dispute data will thus require a resource such as the I2WD Geospatial Ontology (http://milportal.org), incorporating also references to the different kinds of fiat geopolitical entities described in [3].

In the case of claims that nations stake upon disputed territory, the information artifacts involved may be descriptive or directive. For instance, during the Six Day War in 1967 Israeli forces seized East Jerusalem and asserted that this region is (and always had been) part of the geopolitical region of Israel. Thus, Israel's claim on East Jerusalem is formulated as a piece of descriptive information. However, this claim was associated with directive information specifying how persons should conduct themselves with respect to that territory, declaring inhabitants of East Jerusalem subject to Israeli law and restricting access to the region by non-Israelis.

Territorial claims are often bolstered by arguments aimed at establishing their truth. For instance, to justify the seizure of Palestinian territories, Geula Cohen of the Israeli Parliament argued in 1999, "The Jews did not come back to Israel to be safe but to build a nation on the lands given to us by the Bible." [4] In this statement, Cohen gives an argument in support of Israel's territorial claim on the basis of divine right. Here, we can distinguish Israel's territorial claim, Cohen's argument in favor of this claim, and the religious beliefs underpinning this argument, all of which are salient to representing the territorial dispute as a whole. Or consider also the 1994 argument of Stephen N. Schwebel, in favor of the legality of the Israeli settlements on the basis of the principle of a sovereignty vacuum [5] (See Supplementary Material Figure 5). Another important feature of territorial disputes are the motives of the disputing parties. Salient motives include:

- total autonomy, independence, or secession (for example of Quebec from Canada, of Scotland from the United Kingdom, of Catalonia from Spain);
- local autonomy (of Kashmir, South Tyrol, Sicily) within one or another existing sovereign nation;
- economic advantage via the exploitation of natural resources (for example petroleum and natural gas in the areas surrounding Hans Island and Paracel Islands, and in the Aegean Sea);

- restoration of territory held to have been lost (of Gibraltar to Spain, of Ceuta to Morocco, of Belize to Guatemala);
- expansion of territory (of Canada, Russia, the United States, Denmark, and Norway into the Arctic Circle);
- strategic gain (of Tuzla Island and the Strait of Kerch for Russia).

This collection of data relating to motives is complicated by the fact that overt motives may differ significantly from hidden ones. We contend in what follows that an ontology of territorial disputes should have the resources to represent motives of both types, though we recognize that obvious problems arise in regard to the latter since it is possible for the motive that is driving a territorial claim to be perfectly concealed. This however is not different in principle from what is involved when gaps in scientific knowledge are revealed by new discoveries.

V. Disputes, Conflicts, and Relational Qualities

Persons, organizations, and governments engage in a wide array of disputes over plans, goals, predictions, decisions, policies, laws, beliefs, property, and territory. Additionally, they engage in a similarly wide array of conflicts. But what kind of things *are* disputes and conflicts, and how do they relate to entities of other sorts?

As a provisional characterization, both disputes and conflicts – whether or not they are territorial in nature – are *relational processes* in BFO terms; thus they are processes dependent upon and involving as participants at least two agents. In the case of a *dispute*, the participants have views that clash, and the dispute unfolds in a series of sub-processes in which each disputant offers claims and arguments in the hope of convincing others of the correctness of their views. A *conflict*, on the other hand, is a relational process that is made up of inherently hostile (violent) interactions among participants. Not every dispute engenders conflict: a disagreement over the precise demarcation of a border, for example, may be resolved through negotiation. And not every conflict involves a dispute, as when one person strikes another in some random attack.

When two agents – which may be either single persons or more or less formally organized groups of persons – are involved in a dispute, then there exists also a *relational quality*, which in BFO terms is a specifically dependent continuant that inheres in multiple bearers and which connects them together. We can refer to this relational quality as a *state of dispute*. A state of dispute comes into existence at a certain time (for example as the result of an act by one of the parties of staking or contesting a territorial claim); but it exists thereafter in its own right until, perhaps through some further act, it goes out of existence. A state of dispute is essentially relational; thus it not reducible to non-relational qualities inhering separately in the involved parties (compare in this respect relational qualities such as claims and obligations).

Once the state of dispute comes into existence and it preserves its identity for as long as it exists even while undergoing a variety of different sorts of changes (for example in intensity of associated conflict). Even while we do not believe that it is possible to articulate an exact account of the identity conditions for disputes over time, we nonetheless believe that it is in many cases unproblematic to identify a given state of dispute as one and the same from one time to the next. Just as an obligation comes into being upon the making of a promise – for example as documented in a written contract – and only ceases to exist upon either the fulfillment of the promise or the waiving of the obligation by the one to whom the promise was made, so a dispute comes into being upon the act of instigation and only ceases to exist upon either (1) a resolution amenable to the parties involved, (2) the ceasing to exist of one or more of these parties, (3) the involvement of further parties for example in imposing a resolution by force, or in creating conditions which deprive the original dispute of its basis.

Consider, now, the special case of territorial disputes, a typical case of which involves two or more governments in dispute over sovereignty in regard to some specific territory. In our view, the state of dispute is an entity existing through time as an entity in its own right inhering in the participant agents. The state of dispute begins to exist because of actions on the part of one of the parties involved (for instance, Nation A moves armed forces into a territory claimed by Nation B; Nation A releases a map that depicts a region that is claimed by B as falling within the geopolitical boundaries of A). Dispute processes may then ensue, for example on the diplomatic level, but the state of dispute exists even during periods of time when no such processes are occurring.

In some cases, such dispute processes lead to outright conflict, and there then arises a new relational quality called a state of conflict, with subtypes including a state of war, between them. Both states of dispute and states of conflict are relational qualities that inhere in two or more agents.

VI. Data about Territorial Claims

A. Kinds of Data

We turn now to the handling of data about territorial claims. Most of these data are readily treated with the resources of the I2WD ontology framework, including the Information Artifact Ontology, Geospatial Ontology, Time Ontology, Event Ontology, and Agent Ontology, with BFO as foundation. This suite of ontologies can be used as a tool for tagging different types of data salient to territorial disputes, including:

(1) Map-based data: representations of geospatial regions in proclamations and agreements between nations, for example as *claimed*, *disputed, demilitarized*, and so on).

(2) Causes of a dispute: assertions relating to the history of a region (especially its political history), statements of motives for claims (concerning natural resources, strategic position, purported loyalties of a population, and so on), incidents prompting the emergence of the dispute.

(3) Arguments: the arguments offered on each side for the legitimacy of a territorial claim can come in a variety of forms, including appeals to international courts, historical documemnts, results of referenda, and geographic and oceanographic reports; they can be presented as official

announcements, through political speeches, diplomatic notes, and so on.

(4) Treaties: Treaties are relational qualities in the sense outlined above. The treaty document serves multiple functions. First, it documents the deliberative process which led to an understanding on the part of the parties involved that the dispute should be brought to an end. Second, it documents the terms of this understanding, which amount to a set of bi-directional obligations to act henceforth in accordance with these terms. Third (at least in the ideal case), by being ratified by the representatives of the involved parties it serves to bring the dispute to an end. And fourth, by containing signatures of these representatives it documents their acknowledgement of these terms and their acceptance of the concomitant obligations. Legal appeals to treaties can thus make reference to both the treaty itself (the relational quality) and to the signed document (an information artifact). Moreover, as we shall see in more detail below, already existing treaties may be utilized as evidence in favor of new territorial claims.

B. False and Disputed Information

Any territorial dispute will involve conflicting information about the status of some territory, the location of the relevant borders, the soundness of the arguments supporting territorial claims, the (descriptive or rhetorical) character of given press releases, the validity of existing treaties and rights of other nations or groups, the interpretation of salient judgments of international law, the history of the disputed region, and so forth. Sometimes the salient information can be classified either as true or false *simpliciter* – for example, in the case of geographical coordinates of given landmarks. In most cases, however, we shall need to refer to claims as true or false in the eyes of one or other of the disputed parties, or as being such that their truth or falsehood is uncertain (with various modalities). We now suggest a way of tagging information along these lines, distinguishing three categories of information that fall short of being true *simpliciter*:

a. Information that has a truth-value that is to a degree uncertain.

b. Information that is not false, but has some related defect, for instance, in being misleading.

c. Information that falsely asserts that a relation or a particular exists when it does not.

Categories a. and b., which cover many territorial claims, can easily be handled within our framework, For instance, it is (currently) uncertain whether Hans Island belongs (or should belong) to Canada or to Denmark, whether Bethlehem belongs (or should belong) to Israel or to Palestine, and whether the Paracel Islands belong (or should belong) to China, to Taiwan, or to Vietnam. Confidence in such claims begins as a cognitive process of assessment that has as input, the claim, and, as output some degree of confidence (*uncertain, very uncertain*, and so on) that will be used to tag the information in our knowledgebase. Information in category b. that is not false, but in the vicinity thereof – because it is metaphorical, bullshit, rhetorically embroidered, evokes codes only understood by its intended audience, and so on – can be handled by tagging the claim as output of one or other kind of performative act (*of misleading*, *provoking*, *inciting*, and so on). In this way, we can draw attention to the fact that the information is being communicated with a special purpose or in a special context that modifies the literal meaning of the words being used.

However, category c. cannot be dealt with so easily. In many cases, analysts do know with a high degree of certainty that a claim is false. For example, in III we noted that Russia has claimed 1.2 million square kilometers of the Arctic Ocean on the basis of a false report of oceanographic measurements of its continental shelf. Our ontology must be able to represent what that report is about, in this case, the 1.2 million square miles that (Russia claims) comprise their continental shelf. The problem is that, on the supposition that the Russian claim is false, there is no such entity as the Russian continental shelf of 1.2 million square kilometers.

A common strategy for representing false statements involves employing reified RDF triples, where an individual RDF statement may be annotated with the quality "false." This strategy allows knowledge about an RDF triple to be expressed in two steps: the first consists in representing the triple by an instance of a statement that has subject, predicate, and object indicated separately in three different triples. The second step involves creating assertions about that instance as if it is a statement – in our case, an assertion of falsehood. This strategy allows for making statements about statements, but it has largely been found to be inefficient by many users, who find that it dramatically increases the run-time of queries – often making them impossible (though this may change in the future with the introduction of new strategies [6].

We are exploring an alternative two-step approach that begins by appealing to the family of *lacks* relations introduced in [7] in the context of a treatment of negative assertions concerning medical documents. For example, the proposed relation *lacks_part* would hold between a particular *p* and a universal *U* whenever *p* has no instance of *U* as part (such assertions will be made where there is an assumption that *p should have* or *is expected to have* a part of this sort, as for example in: *John is missing his left arm*. Since the particular (John) and the universal (*left arm*) both exist, the assertion of a *lacks_part* relation between them is perfectly in order from a realist point of view. When applied to territorial disputes, this strategy would allow us to posit the fact that corresponds to a false claim: e.g. that the Russian continental shelf *lacks* an extension covering 1.2 million square miles. We can then tag the false claims as being both *false* and also *about* the corresponding lacks relation. This allows us to interpret the class of false information content entities as bearing an *is about* relation to an existing portion of reality. This allows us to deal with false believes held for example by specific governmental organizations while remaining in conformity with the principles of ontological realism.

VII. ARGUMENTS, MOTIVES, OBJECTIVES

Another significant component of territorial claims are the *arguments* made by disputing parties in defense of their claims. Any adequate representation of territorial disputes needs to capture the arguments for or against the truth of given claims, together with the other information content entities delineated above. To this end, we likewise treat arguments as information content entities borne by documents of a range of different sorts. Like claims, arguments are generically dependent continuants that can exist in many bearers, for example in multiple hard drives, in printed form in newspapers, in fliers posted on walls, and so forth.

We should think of arguments as wholes that are comprised of informational parts. For example, the argument supporting the thesis that Russia exercises sovereignty over half the Arctic Ocean involves both the *conclusion* that Russia exercises sovereignty over a particular region and other claims (the *premises* of the argument) offered to support that conclusion, here: that international law stipulates the criteria for a nation's continental shelf, that the measurements of Russia's continental shelf include half the Arctic Ocean, and that whatever falls within a nation's continental shelf belongs to that nation. The argument in favor of Russia's sovereignty over half the Arctic is the logically ordered collection of these information artifacts. Dividing an argument into its parts allows our ontology to keep track of the ways arguments are amended over time. In the first place, we can tag the claim that Russia exercises sovereignty over the Arctic as the conclusion of an argument. We can then situate that claim within an aggregate of other relevant claims. If different reasons are given at a different time, then we can treat those supporting reasons as comprising a separate argument on behalf of the same conclusion. Finally, if some argument is bolstered or diminished by new evidence (e.g., if an independent party issues the results of new and more precise measurements of Russia's continental shelf) then we can represent that emendation.

Next, we consider the various kinds of arguments that could be offered in support of different kinds of claims. Kinds of argument are differentiated by what kinds of considerations they appeal to, for example evidence from geography, from geology, from history, and so on. Brian Sumner [8] identifies nine such kinds of considerations nations might appeal to in defense of a given territorial claim:

(1) Treaty Law: Treaties between nations form the basis of a strong legal appeal for the legitimacy of a claim of sovereignty over given territory. However, these treaties are also disputable, most importantly by third parties, who were not included in the making of the treaty, but who have other considerations in favor of a claim over the territory in question. Further, treaties may turn out to expire or suffer revocation at a later date, e.g., if colonizers of some land made a treaty concerning that land's territorial borders.

(2) Geography: Geographic and environmental features naturally suggest territorial boundaries. In Sumner's words, "Mountain ranges, rivers, oceans, and other bodies of water and physical formations have perennially separated political entities." Our ontology readily accommodates the use of geographic features as evidence for a territorial claim, insofar as it differentiates between the geographic features themselves (mountains, oceans, etc.) and the geopolitical boundaries that are claimed to coincide with these geographic features. (Recall that these claims may be tagged as false or disputed.)

(3) Economy: An economic argument in favor of some territorial claim makes appeal to economic necessity. In such cases, a nation claims that the territory in question is necessary for its sustenance or development. Such claims may include appeal to the necessity of sea-routes, aerial routes, trading establishments, natural resources, raw materials, agricultural potential, or foreign investment for a nation's flourishing. (Such arguments may also make reference to a nation's proximity to certain economically valuable resources, but strictly these considerations are adjudicated by UNCLOS, and do not flow from the existence of economically valuable considerations taken in and of themselves.) Assessment of an argument from economic necessity requires assessment of its individual components. Our ontology therefore captures information pertaining to, for example, deposits of natural resources, trade and transport routes, their economic value to the territory in question, as separately evaluable entities.

(4) Culture: An argument from culture appeals to "common language, religion, kinship, or other cultural characteristic that defines the group of people living in a particular territory." Quebec's attempted secessions from Canada have involved arguments of this sort, factors relating common cultural background being offered as evidence for the drawing of new territorial borders insofar as these factors would contribute to the unification of a region's population. In some territories, religion plays a strong unifying role. Again, our ontology captures such arguments by treating cultural factors such as ethnicity, religion, and language as separate entities.

(5) Effective Control: Arguments from effective control appeal to facts about a nation's *de facto* uncontested administration of a given territory. Historical appeals to such administration are used to support arguments for adverse possession of a region. Analogous arguments are applied, too, over longer time scales, for example in support of native populations' claims to sovereignty over regions of territory over which they once held sway and which have subsequently colonized by outsiders.

(6) History: Many territorial claims make reference to purported historical facts about the region under dispute. For instance, China claims that their fishermen have made use of the bulk of the South China Sea for centuries, and that this fact is strong evidence in favor of a claim over the vast majority of the Sea. (This also shows the overlap between historical and economic considerations, insofar as China's claim rests upon the longstanding economic importance of the region to China.)

(7) Uti Possidetis (meaning "as you possess") is a principle upon which newly independent nations inherit the boundaries determined by colonial powers. Nowadays, this principle is only rarely invoked. More to the point, it is usually taken to be relatively weak evidence for a claim, and considerations (1)-(6) generally take precedence over *uti possidetis*.

(8) Elitism: Arguments under this heading comprise involve appeal to the fact that one participant in a territorial dispute is in one or other respect in a superior position with respect to another participant. This includes appeals to divine right, the superiority of one's civilization, or racial superiority. Such claims, too, have become increasingly rare, and arguments from elitism are nowadays considered to be

relatively weak. However, that is not to say that such arguments are not made. Recall Cohen's claim, quoted earlier, that the Jewish people returned to Israel "to build a nation on the lands given to us by the Bible."

(9) Ideology: Finally, arguments from ideology make reference to ideological factors for the legitimacy of some territorial claim. Sumner cites anti-colonialism and the movement for social justice as sources of ideological arguments for territorial claims.

This classification becomes especially useful when we analyze arguments employed in territorial disputes from the perspective of the known or suspected *motives* of the governments involved. For instance, there is reason to believe that the Arctic region is the site of untapped natural resources. Russia's stated arguments in favor of its sovereignty over half this region may make no mention of these resources, and yet an analyst can reasonably suppose that the intention to exploit those resources is one of Russia's motives for claiming sovereignty. Thus, we first propose distinguishing between stated arguments and known or suspected motives, and then viewing arguments in light of these motives. Motives comprise the objectives the government has in winning the territorial dispute, whereas arguments are devices to facilitate progress toward gaining these objectives. That is to say, arguments are only one part of a government's plan to realize its objective, which is authority or sovereignty over a given disputed territory.

This objective will in every case be embedded within a nation's efforts to realize broader economic, political, ideological, and military goals. A country might have a plan whose objective is to grow its economy, and this plan might include subplans for some sort of political or military action to achieve sovereignty over some region and exploit its resources. ([9] provides a detailed discussion of some Norwegian, Russian, and Finnish strategies pertaining to territorial claims in the Arctic along these lines.) The components of these subplans will in turn involve, at still lower levels, plans concerning how to achieve this task, whether by vigorously defending some claim at the United Nations or intimidating the military craft of other nations in a given area.

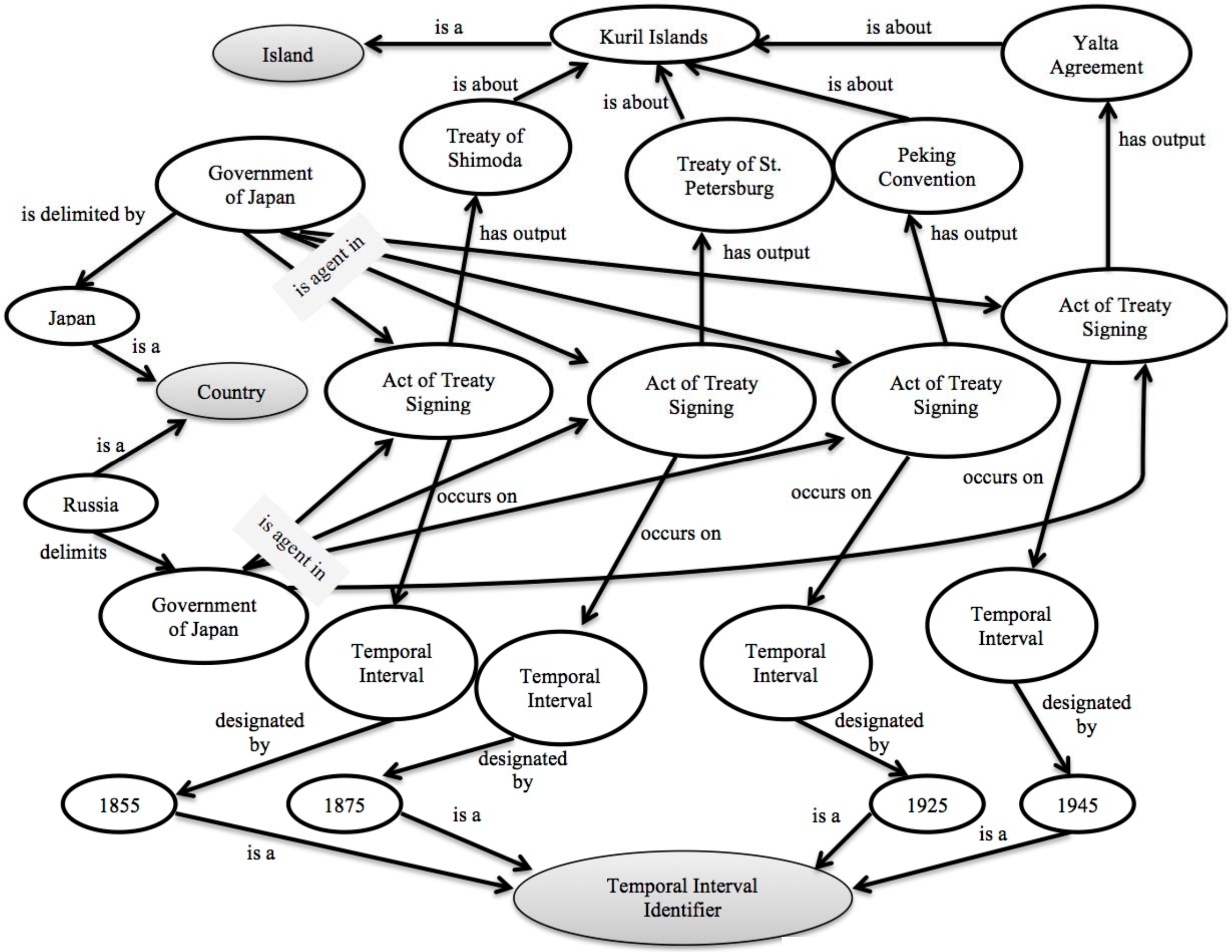


Figure 1: Kuril Island Treaty Timeline

VIII. APPLICATIONS: THE CASE OF THE KURIL DISPUTE

We will now apply our ontological approach to data about a specific dispute, namely that between Russia and Japan over the Kuril Islands, which has been on-going for over a century (Bobic, 2012, see Figure 1). In 1855, Russia and Japan signed the Treaty of Shimoda, which divided the northern and southern islands between them. One island, named Sakhalin, contained Russian, Japanese, and Ainu inhabitants, and so was omitted from the Treaty. Hostilities increased among the inhabitants of Sakhalin, and in 1875 Russia and Japan signed the Treaty of St. Petersburg, which gave sovereignty of Sakhalin to Russia and sovereignty of all the remaining Kuril Islands to Japan. As Bobic notes, the interest in the Kuril Islands is largely strategic. The islands have held strategic significance especially for Soviet Russia, which sought to station submarines in the area. Their economic value, however, is meager, as there are few petroleum or mineral deposits. There is some possibility of oil and gas reserves, but the amount is unknown. Finally, the islands have symbolic significance, insofar as they have been the site of important violent struggles between Russia and Japan. As Bobic reports, "the symbolic value of the islands matters the most to the local Russian residents, who believe that this was the land won with the blood of Russian soldiers." [10] Early in the twentieth century, the dispute escalated into full conflict between Russia and Japan, which was eventually resolved through ratification of a peace treaty through the mediation of the United States. After the October Revolution of 1917, Russian forces again clashed violently with Japanese in the region, but this eventually led to another agreement, the Peking Convention of 1925. Following the conclusion of World War II, Stalin expressed his desire to seize the Kuril and Sakhalin Islands from Japan, and did so with Roosevelt's blessing in the Yalta Agreement of 1945 (see Figure 2). Japanese-Russian relations were "normalized" in 1955, but the dispute over the islands remained. Late in the 1970s, the Soviet Union stationed troops on some of the islands, and a few years later Japan sent Prime Minister Suzuki to visit the southern islands in the archipelago, and designated a "Northern Territories Day," which only served to escalate tensions. At last, starting in 1990, Yeltsin in Russia moved toward a proper resolution of the dispute, and met with Japanese officials in 1993. Eventually, this led to an agreement of mutual use of fisheries in the region and of visa-free travel for Japanese to the area. Most recently Putin, however, has stalled further talks on resolution of the dispute. (See Figure 3 in the Supplementary Data provided at http://ontology.buffalo.edu/14/territorial-disputes/.)

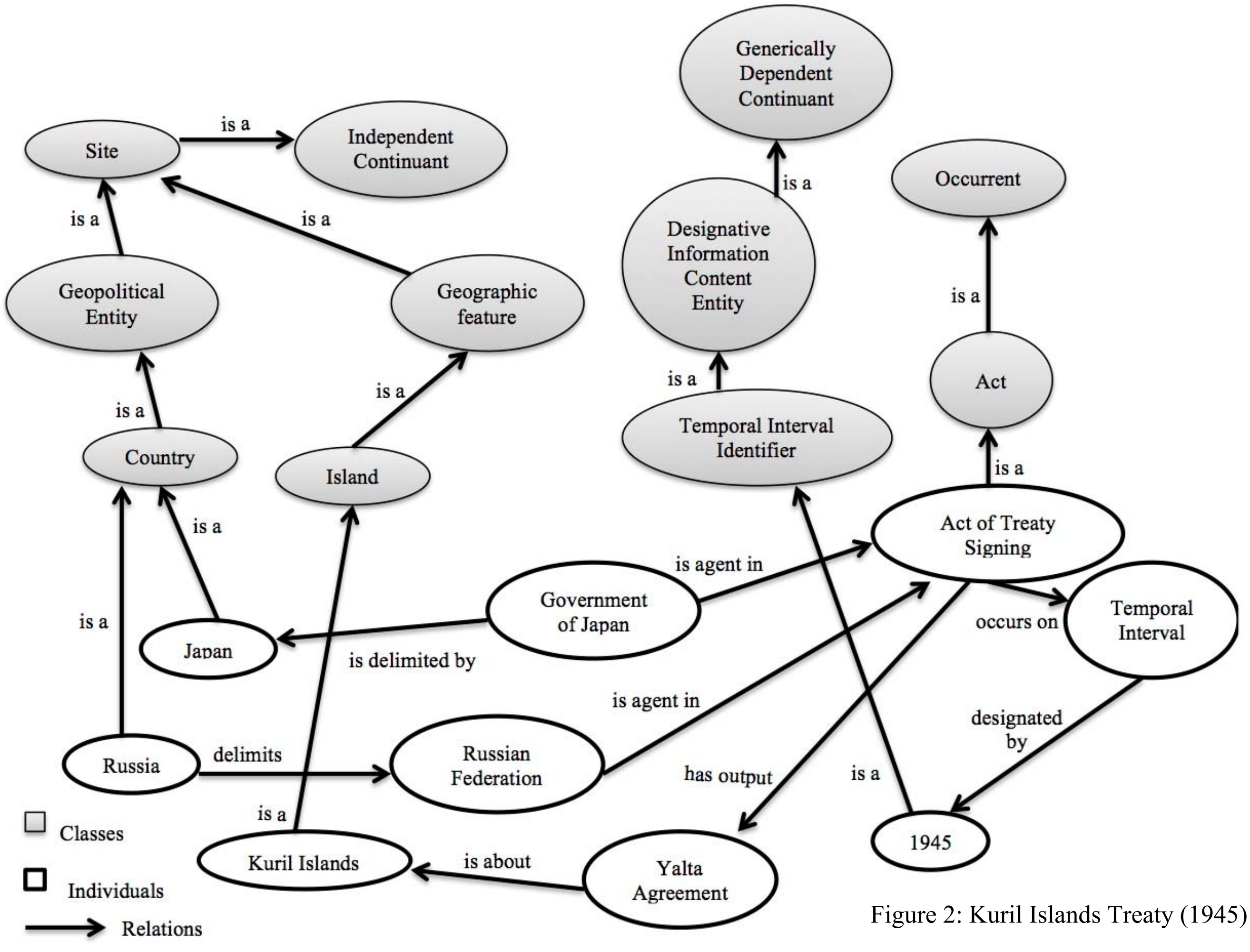


Figure 2: Kuril Islands Treaty (1945)

Our ontological approach is capable of representing this complex situation, spanning over a century of tension, conflict, treaties, and shifting borders. We do this by first considering the timeline formed by a sequence of acts of treaty signing. Each such act has the output of a signed treaty, and it also occurs at a specific temporal interval, which can be timestamped. Thus, there is an act of treaty signing with output the Treaty of St. Petersburg. This act has two participants – the governments of Japan and Russia – and occurs at a temporal interval that is designated '1875'. Such temporal intervals are proper parts of the temporal interval occupied by the territorial dispute taken as a whole. Two discrete increases in conflict, resulting in armed conflict, occur on their own temporal intervals, designated by dates, and each overlaps with parts of the temporal interval of the dispute. (See Figure 2 figures in accompanying material.)

In addition, the symbolic importance of the islands rests on an instance of the disposition type we have labeled *nationalism*, and this symbolic importance serves as an *ideological motive* for the dispute. The *objectives* of the governments, on the other hand, turned on *strategic naval advantage*. Both motives and objectives are distinct from the explicit arguments put forward by participants in the dispute at different times through the century. Acts with arguments as outputs can be represented in their turn as occurring on specific temporal intervals that are designated by particular dates.

IX. Conclusion

We have surveyed the ways in which our ontological approach can capture the features of a territorial dispute within the framework of the Basic Formal Ontology by appealing to the I2WD ontology suite. In particular, the ontology we propose offers the ability to capture the peculiar character of disputes and associated conflicts, it has a strategy to deal with both false and disputed information, and with the various kinds of arguments, motives, and objectives at work within them. In addition to terms representing entities such as claims, arguments, territories, and roles, the ontology must specify also the relationships among the diverse elements involved, for example, the relationship between the content of a piece of propaganda and an objective, or between an argument and a claim, relations such as *aboutness*, *support*, *ratified by*, and so on. Some of these relations are illustrated in the Figures.

Acknowledgements

We wish to thank John Beverley for incisive comments on an early draft of this paper, and Ron Rudnicki and Yonatan Schreiber for fruitful discussion of these issues.

Supplementary Data

Provided at http://ontology.buffalo.edu/14/territorial-disputes/:

Figure 1: Kuril Island Treaty Timeline
Figure 2: Kuril Islands Treaty (1945)
Figure 3: Russian Troops FDICE
Figure 4: Arctic Dispute
Figure 5: Schwebel Argument
Figure 6: North Korea FDICE

References

[1] Barry Smith and John Searle. 2003. "The Construction of Social Reality: An Exchange," *American Journal of Economics and Sociology* 62, 2: 285-309.

[2] Kathryn Isted. 2009. "Sovereignty in the Arctic: An Analysis of Territorial Disputes & Environmental Policy Considerations," *Journal of Transnational Law & Policy* 18, 2: 343-76.

[3] Barry Smith. 1995 "On Drawing Lines on a Map," in A. U. Frank and W. Kuhn (eds.), *Spatial Information Theory. A Theoretical Basis for GIS*, Berlin and Heidelberg: Springer-Verlag, 475-84.

[4] Milton Viorst. *Sands of Sorrow: Israel's Journey from Independence* (I.B. Tauris, 1987).

[5] Stephen M. Schwebel. 1994. *Justice in International Law: Selected Writings (What Weight to Conquest?)*. Cambridge: Cambridge University Press. pp. 521–526.

[6] Vinh Nguyen, Olivier Bodenreider, and Amit Sheth. "Don't like RDF Reificiation? Making Statements about Statements Using Singleton Property." *Proceedings of the 23rd International World Wide Web Conference 2014*: 759-770.

[7] Werner Ceusters, Peter Elkin, and Barry Smith. "Negative Findings in Electronic Health Records and Biomedical Ontologies: A Realist Approach." *International Journal of Medical Informatics*, 2007 December; 76 (Supplement 3): 326-333.

[8] Brian Taylor Sumner. 2004. "Territorial Disputes at the International Court of Justice," *Duke Law Journal* 53: 1779-1812.

[9] Stefano De Luca. 2013. "The Cartographic Reasoning in the Arctic: Modern Territorial Representations of the State in the Arctic Strategies of Norway, Russia and Finald." Master's Thesis, University of Tampere.

[10] Marinko Bobic. 2012. "Words or Swords: Russia's Strategies in Handling its Territorial Disputes." Master's Thesis, Leiden University.

[11] Anneta Lytvynenko. 2011. "Arctic Sovereignty Policy Review," URL: http://www4.carleton.ca/cifp/app/serve.php/1355.pdf. Accessed August 25, 2014.

[12] Marsha Walton. 2009. "Countries in tug-of-war over Arctic resources," *CNN* (January 2, 2009), URL: http://www.cnn.com/2009/TECH/science/01/02/arctic.rights.dispute/. Accessed August 25, 2014.

Ontological Support for Living Plan Specification, Execution and Evaluation

Erik Thomsen
Charles River Analytics
Cambridge, MA
ethomsen@cra.com

Frederick Reed
Charles River Analytics
Cambridge, MA
freed@cra.com

William Duncan
NCOR
Buffalo, NY
wdduncan@gmail.com

Tatiana Malyuta
New York City College of Technology, NY, NY
tmalyuta@thedatascience.org

Barry Smith
NCOR
Buffalo, NY
phismith@buffalo.edu

***Abstract*—Maintaining systems of military plans is critical for military effectiveness, but is also challenging. Plans will become obsolete as the world diverges from the assumptions on which they rest. If too many ad hoc changes are made to intermeshed plans, the ensemble may no longer lead to well-synchronized and coordinated operations, resulting in the system of plans becoming itself incoherent. We describe in what follows an Adaptive Planning process that we are developing on behalf of the Air Force Research Laboratory (Rome) with the goal of addressing problems of these sorts through cyclical collaborative plan review and maintenance. The interactions of world state, blue force status and associated plans are too complex for manual adaptive processes, and computer-aided plan review and maintenance is thus indispensable. We argue that appropriate semantic technology can 1) provide richer representation of plan-related data and semantics, 2) allow for flexible, non-disruptive, agile, scalable, and coordinated changes in plans, and 3) support more intelligent analytical querying of plan-related data.**

Keywords—adaptive planning; outcomes assessment; ontology

I. The Need for Adaptive Planning

"No plan survives first contact with the enemy" (Clausewitz, *On War*). Real world uncertainties all but guarantee that even the most carefully developed plan will not be carried out exactly as intended. The military response, as in the business domain, has been to increase the speed and agility of planning and execution [1-4]. On the strategic level, the transition from the Joint Operation Planning and Execution System (JOPES) to an Adaptive Planning and Execution (APEX) system exemplifies this trend. In addition to speeding up the deliberate planning and review cycle, these efforts seek to increase the number of planned options and contingencies.

According to the Adaptive Roadmap II, signed by the Secretary of Defense in March 2008, the ultimate goal is to provide plans that are "maintained continuously within a collaborative environment" to reflect any changes that impact any significant aspects of a plan. Such plans will together form something the Adaptive Roadmap calls a "living plan." Plans may need to be adjusted to maintain their relevance based on changes in the world (e.g., weather, location of enemy troops, troop readiness, air assets). Additionally, they may need to be adjusted in order to maintain their coherence within a system of plans, such as when the goals of supporting or supported plans change.

II. The Idea of the Living Plan

In the current state of military planning – as encapsulated in Joint Doctrine (JP 5.0) – a distinction is drawn between deliberate planning and crisis action planning. Deliberate planning is supply driven. Plans are static information objects created as the outputs of a deliberative, rule-governed process, and stored in a repository until needed. They may be created years ahead of actual use, or they may never be used at all. Crisis actions plans are demand driven: something happened and we need an urgent response; because the response should involve a degree of organized action, planning is needed. Crisis action planning is a response to the uncertainty involved in our knowledge of real-world states. But even deliberate planning rests on an institutional acknowledgement of our inability to accurately predict the future, in that Doctrine allows the making of ad hoc resource requests which deviate from the deliberate plan as specified. Sometimes, on first contact with the enemy, deliberate plans break and workarounds are needed. Regardless of the quality of the prior deliberation that went into the deliberate plan, the need for such corrective actions as a result of the unanticipated interactions between blue forces and the world make for suboptimal procedures.

The goal of the living plan is to remove this 'breaks because it would not bend' feature of the deliberative plan by minimizing the distinction between deliberate planning and crisis action planning through a new type of planning process that is marked by constant update in light of updates in our real-world knowledge. The idea is to embed into the very fabric of plan representation our uncertainties about the world, so that the activity of planning is transformed from one of the creation of plans as outputs to a process of continuous plan development. The living plan itself becomes a probabilistic, branching information artifact – a representation of the moment-to-moment intentions not merely of single platoon commanders but of the military as a whole. It incorporates at each phase representations of multiple alternative courses of action which are continuously changing in light of actual and projected states of the world, adjacent plans, supporting and supported plans.

III. Adaptive Planning Requirements

We believe that any computational approach to supporting the Secretary of Defense's goal for living plans must meet six critical requirements.

First, it must be able to represent all the types of entities and relationships, knowledge about which is important to maintaining a living plan. This requires a *highly expressive representational capability* to capture, manage, and reason over plans, plan elements (e.g., goals, available assets, weather, battle terrain), and their relations within a system of plans.

Second, any approach must be able to detect meaningful changes that impact plan relevance and coherence. This requires *effective monitoring and sensitivity analysis* to identify in a reliable and scalable way those changes which are of significance to the system of plans [5,6]. Recognition of the significant changes must then trigger processes that maintain the relevance and coherence of this system at multiple levels and across plan elements.

Third, any approach requires *coordinated adjustment processes*, which are needed to fulfill the second requirement (above). Such processes must be able to run independently, be applicable (when necessary) to real-time conditions, and be capable of harmonizing with other large-scale plan adjustments.

Fourth, any such approach requires *automated information extraction and routing* because maintaining realistic plans requires more information processing than can be achieved through manual methods alone.

Fifth, whether in support of human planners, warfighters during mission execution, operations assessment staff, or automated systems performing the same tasks, any approach needs to *support analytical queries against the ensemble of plan-related data*. Since plan-related data is very heterogeneous, this amounts to applying a unified structured query front end to structured and unstructured data on the backend.

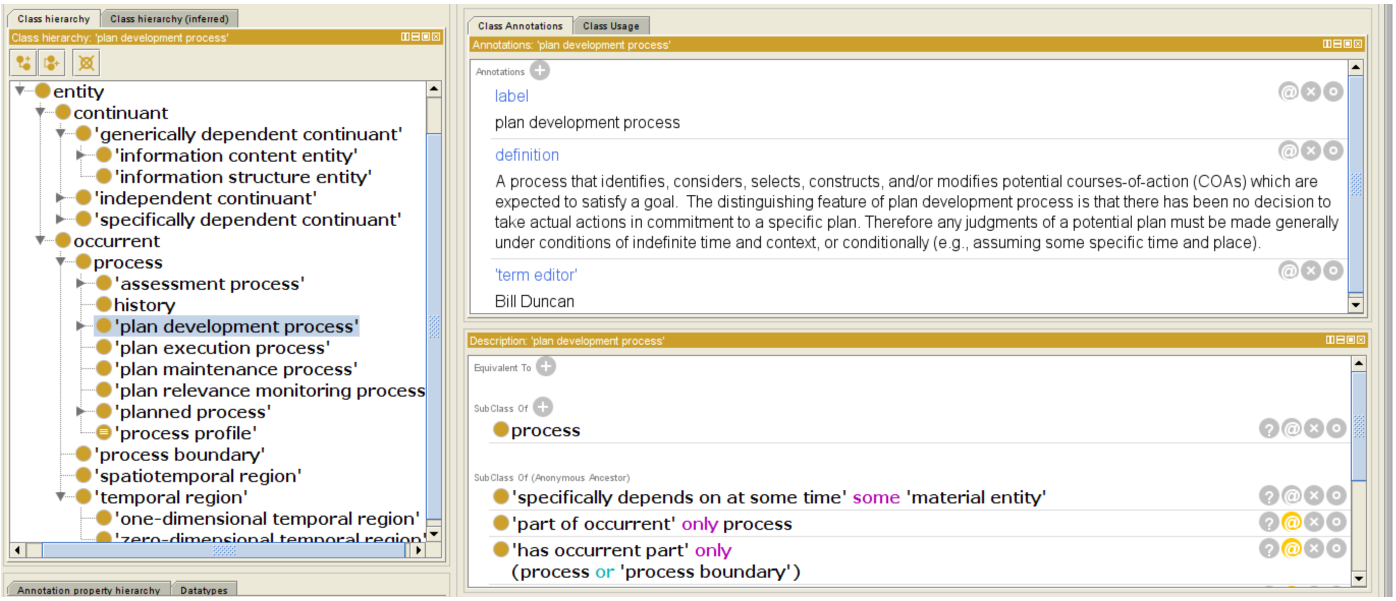
Sixth, joint warfighters at all levels of command will need to collaboratively plan and execute in conjunction with semi-automated adaptive planning systems. Therefore, any approach for providing living plans must support *extensible and versatile interactive applications* that can deal with the sorts of diverse but integrated user environments required for living plans.

Relative to the six requirements described above for supporting the Secretary of Defense's goal for living plans, our overall approach is based on the idea that semantic representation of data by means of ontologies, combined with probabilistic classifiers operating in a transactional environment, will allow the needed representation, monitoring, analysis, sharing and querying of information at distinct levels of granularity and detail and across distinct applications. The system will be required, for example, to display a JFACC's view of ATO mission plans, a squadron Commander's view of the day's mission plan, and STRATCOM's view of a Theater. As in other domains, the semantic approach is designed to reduce information siloes, and enable effective tailoring of knowledge and information to different needs. It is designed also in such a way as to allow incremental improvements over time, as shortcomings in the framework uncovered at any given stage are rectified in subsequent stages.

In what follows we focus on the first and fifth requirements described above: for rich representations of data and semantics, and for the capacity to use such representations in mounting queries against plan-related data.

As regards the former, we describe the coverage domain of our proposed Plan Ontology (see Figure 2) in terms of how we: (a) model plans in terms of cyclical phase-specific attributes; (b) embed metrics that relate plans to world conditions; and (c) embed meta-metrics that use the metrics under (b) to create an incremental plan and plan-execution improvement process across the whole system. On each level multiple families of related terms will be required, including definitions and axioms specifying the relations between them.

As regards the latter, we describe how queries are passed through parts of the system in order to illustrate some of the semantic relations that need to be computed in order to support analytically useful queries over living plan data.



Class hierarchy
Class hierarchy (inferred)
Class hierarchy: 'plan development process'
Annotation property hierarchy
Datatypes
Class Annotations
Class Usage
Annotations: 'plan development process'
Description: 'plan development process'

- entity
  - continuant
    - 'generically dependent continuant'
      - 'information content entity'
      - 'information structure entity'
    - 'independent continuant'
    - 'specifically dependent continuant'
  - occurrent
    - process
      - 'assessment process'
      - history
      - 'plan development process'
      - 'plan execution process'
      - 'plan maintenance process'
      - 'plan relevance monitoring process
      - 'planned process'
      - 'process profile'
    - 'process boundary'
    - 'spatiotemporal region'
    - 'temporal region'
      - 'one-dimensional temporal region'
      - 'zero-dimensional temporal region'

Annotations

label

plan development process

definition

A process that identifies, considers, selects, constructs, and/or modifies potential courses-of-action (COAs) which are expected to satisfy a goal. The distinguishing feature of plan development process is that there has been no decision to take actual actions in commitment to a specific plan. Therefore any judgments of a potential plan must be made generally under conditions of indefinite time and context, or conditionally (e.g., assuming some specific time and place).

'term editor'

Bill Duncan

Equivalent To

SubClass Of

process

SubClass Of (Anonymous Ancestor)

'specifically depends on at some time' some 'material entity'

'part of occurrent' only process

'has occurrent part' only
(process or 'process boundary')

Figure 1: Fragment of the draft Plan Ontology at http://ncor.buffalo.edu/plan-ontology

IV. Representing plans in Recurring Phases

To better understand and support the notion of continuous, living plans, we require a view of planning that is more abstract than is traditionally employed. The simplistic notion of 'the plan' created prior to 'the execution' is at odds with our view of planning as a dynamic, continuous, iterative process that not only adapts to the effects of planned actions, but also adapts the process of planning itself in ways designed to achieve more satisfactory outcomes over time.

Our model focuses on three primary factors in the planning process:

1. different phases of the planning process (successive phases within a given course of planning processes),
2. types of judgments within each of those phases that enable effective planning, and
3. information, including metrics, on which these judgments are grounded.

On the traditional view, planning only happens periodically as a precursor to its execution. Here, in contrast, we view the total planning process computationally as forming a series of parallel, interacting courses or flows at a number of different levels. These processes unfold dynamically, with changes in any given course being communicated to parallel and hierarchically related courses wherever changes in the latter are required. The system is organized in such a way that updated versions of needed plans and subplans can be generated at any point in time.

Each parallel course is itself seen as being organized into a succession of three phases corresponding roughly to the first three phases of the well-known Plan Do Check Act (PDCA) cycle, and similar models. A difference is that the phases in our framework are viewed as continuous and intermeshed with each other rather than discreet. Especially the *Act* phase, where adaptive actions are taken, is distributed and continuous across the other phases.

- *development* – This phase consists of processes of identifying, considering, selecting, constructing, and/or modifying potential courses-of-action (COAs) that are expected to satisfy a goal. This includes the process of creating and maintaining potentially executable 'plans sitting on the shelf' in traditional, deliberate planning – referred to in our ontology as 'plan specifications'. The distinguishing feature of this phase is that there has been no decision to take actual actions in conformity with and under commitment to any specific plan.
- *execution* – This phase involves processes of planning while acting according to a particular planned COA. Unlike random or spontaneous actions, such planned processes can be evaluated relative to the plan. For example, indicators can be used to judge whether the intermediate effects of planned actions are consistent with expectations. But, as the plan has not yet terminated, the net effect of all planned actions relative to the goal set forth in the plan cannot be judged. A key planning process in the execution phase is the making of a decision to terminate execution because the goal has been achieved, or because the plan is no longer relevant or coherent, or is being executed unsatisfactorily.
- *post-execution* – This phase involves the post-execution processes of interpreting and judging an executed plan and its outcomes relative to expectations. In this process, all actions taken under commitment to the plan have been taken. Thus their net effect can be assessed relative to the specified goal. The primary purpose of the processes involved in this post-execution phase is to enhance future planning, for example by:
    - o defining new goals;
    - o clarifying existing goals;
    - o improving effectiveness in achieving goals.

Associated with processes of each of the mentioned types are four basic planning-related judgments that enable reasoning aimed at leading to the creation and selection of better plans:

- *relevance* – How well does the current state of planning relate to actual or anticipated external world conditions, such as constraints, opportunities, planned outcomes, unplanned side-effects, etc.?
- *coherence* – How well do the processes of planning on-going in the current phases relate to other synergistic planning processes. In other words, are they in conflict or coherent with other friendly force, coalition, political, etc. planning?
- *planning-assessment* – How well were the processes in each phase of planning performed by the *planner*, from a single person to an organization?
- *meta-metric learning* – How well does the current set of metrics support the goal of evolutionary improvement of the entire planning process (and, as a consequence thereof, the entire process of creating and executing and evaluating plans)?

V. Representing Recurring classes of metrics in support of cyclical plan phases

In this section, we bring together the three factors of planning outlined above – phases, judgments, and metrics – to see how they merge to form a more complete picture of a continuous adaptive planning process. For each combination of planning phase and judgment we provide example metrics. These are provided here for illustrative purposes only, and especially as concerns plan execution our framework will draw on the extensive list of Measures of Effectiveness and Performance identified in salient doctrine for the tasks of the Universal Joint Task List, for example as described at: http://www.dtic.mil/dtic/tr/fulltext/u2/a398683.pdf

A. *Plan Development Phase*

1) *Relevance*

Example metrics informing the judgment whether a potential plan will be relevant to some anticipated world state:

- **Values and locations of relevant adversary assets** (a plan to invade a country to remove WMD stockpiles would be irrelevant if there were no stockpiles, or if existing stockpiles were unreachable in a timely manner)
- **Number of red operational defensive SAM sites** (a plan that did not either act to reduce this number, or account for blue attrition because of them, would not be relevant)
- **Number of blue re-fueling tankers available during a period** (a plan with more missions than could be supported for refueling would not be relevant)
- **Network of adversary command communications** (a plan that intends to cripple communications by taking out a central node is not relevant if the network is decentralized and/or has alternate paths)

2) Coherence

Example metrics informing the judgment of whether a potential plan will be coherent with other related planning:

- **Rates of attrition of shared assets** (a plan that over-optimistically assumes assets will remain available after another plan executes is not coherent)
- **Times of anticipated/actual actions that are signs of intentions** (a plan that assumes an element of surprise is not coherent with another plan that takes earlier actions that signal a shared or related intent)
- **Intentions of non-military planning in Area of Operations** (a military plan that depends on large-scale destruction of economic infrastructure, apparatus of civil authority, etc. is not coherent with a political plan that seeks to rapidly restore civil rest and order)

3) Planning Assessment

Example metrics informing the assessment of planning performance during plan development:

- **Time required to reach plan execution phase** (compared to predicted, needed, historical, and so on)
- **Number of substantially different COAs and embedded options considered** (based on the assumption that the larger the number of options the better is the understanding of the space of options)
- **Number of relevance and coherence metrics considered** (by some definition of *considered* and a procedure for *counting* separate metrics)
- **Length of review chain prior to approval by Commander** (includes first-pass and re-review cycles)

4) Meta-Metric Learning

Example meta-metrics describing how well the relevance, coherence and planning assessment metrics support plan development, and enable improvement of the metrics – and thus of the total planning process – over time. Meta-metric learning often requires data over combinations of planning phases.

- **Inter-phase meta-metrics deriving from correlations between some earlier-phase metric with some later-phase metric relating to outcomes** (for example: if the number of options embedded in COAs has historically correlated positively with post-execution assessment metrics indicating greater satisfaction of plan goals, then it may become a more positive metric that is given greater weight in future plans)
- **Correlation between intra-phase metrics generally considered positive (or negative)** (for example: the number of COA options considered is itself to be viewed as a positive metric; but if this number goes up in such a way that the time required to bring a plan to execution goes up at the same time (which is considered negative), then this suggests an optimization is possible, or perhaps a different metric, such as measuring the difference in time between completing a plan and its estimated time of execution rather than total time)
- **Percent of relevance and coherence metrics with measures above a certain level of belief/confidence** (over time, the confidence in metrics should be driven up, for example the confidence in metrics of adversary state such as number of SAM sites should be actively improved with better sensors and analysis processes)
- **Number of corrections made to a metric** ('corrections' means: substantial changes in a metric which are made on the basis of evidence contradictory to the original estimate of what sort of metric would be needed; for example: contradictory evidence that the current WMD estimate, made by whatever process, is wrong leads to improving the process that led to this estimate).

B. Plan Execution

1) Relevance

Metrics informing the judgment whether an actual plan being executed remains relevant to actual conditions, such as constraints and opportunities:

- **Cloud height over intended target** (may violate constraint of target visibility)
- **Number/rate of adversary unit surrenders or other change in adversary offensive activity** (may indicate plan assumptions regarding adversary's will to fight are incorrect or not relevant)
- **Aggregate Measures of Performance (MOPs) for current actions** (low levels of mission performance may indicate that the pre-conditions and contexts for actual actions were not satisfactorily planned – for example low levels of destroy, degrade, deny, disrupt (4Ds) may indicate poor intelligence, weaponeering, etc.)

2) Coherence

Example metrics informing the judgment whether a plan remains coherent over time:

- **Changes in planned asset availability** committed by other plans (for example: there are assets which the plan assumes other plans do not require)
- **Success rate of synchronization points** (if plans have explicit specifications of COA relationships, defined-execution windows, handoffs, meetings, supporting events, and so on, then what is the rate at which these relations are successfully maintained?)

3) Planning Assessment

Example metrics informing the assessment of whether the plan is being executed satisfactorily:

- **Percent of scheduled missions flown on time** (assessing compliance with plan, not outcomes)
- **Rate COA modifications made per unit time** (a better specified plan might require a lower rate of modifications)
- **Aggregate time delays of actual execution for planned simultaneous actions** (for example in massing fires in planned combined air strike and artillery)
- **Time from a relevant change in world state to the appropriate change in COA** (for example: time from when the new target location information is obtained to time when a new mission tasking has been created that accounts for the new information)

4) Meta-Metric Learning

Meta-metrics describing how well the relevance, coherence and planning assessment metrics support plan execution, and enable improvement of the metrics:

- **Inter-phase metric correlation** (for example: low correlation between missions flown on time and post-execution MOE metrics may suggest that flight promptness is not as important as thought, perhaps because late flights were able to act on better, more recent information)
- **Intra-phase metric correlation** (for example: a negative correlation of rate of COA changes and aggregate time delays of planned simultaneous actions may suggest that allowing more frequent COA changes to constructively maintain coherence is beneficial, notwithstanding the expected disruptive effect of the changes; better metrics might distinguish COA changes by class of initiating event, such as new information, command decision, and so on; as the framework itself becomes more sophisticated in its reasoning power, more frequent COA changes will themselves become more easily accommodated by the planning system)

C. Post Execution

1) Relevance

These are metrics informing the judgment of the effects of the executed plan on world state, particularly relative to intended outcomes. In addition to the more typical post-operations assessment process, there are other ways to conceptualize post-execution relevance. For example: do the lessons drawn from assessment have relevance to the current or future world? Is the originally desired outcome – such as destroying (or building up) another actor's offensive capability (for example arming the Taliban) of continued relevance? Or is it becoming less relevant, for example because they have changed sides?

- **Number of missiles landing in homeland** (this is said to have been the post-execution operations metric for the recent Gaza invasion)
- **Number of computer systems not patched for exploit X** (exploit X might have worked well on this occasion, but if the adversary has since learned about it and therefore patched the prior vulnerability, the simple assessment that it worked well previously is not particularly relevant for future planning)

2) Coherence

Metrics informing the judgment how the net outcome is coherent with other plans (in any phase)

- **Actual asset attrition** (for example: achieving the current plan objective with more or fewer bullets may not matter to the current plan, but it may harm/limit other planning. This is following the notion that Relevance is assessing the relation of the outcome to the *current world state*, so Coherence would be the relation between the outcome and *other plans*.)
- **Degree to which actual net outcome facilitates or limits COAs of future plans** (e.g., confident removal of WMD threat makes other plans easier to develop and execute)

3) Planning Assessment

Metrics informing the judgment of how well the post-execution planning process is performed:

- **The number of indicator metrics integrated into the overall goal assessment** (for example: if goal end-state is to influence future behavior, then more indirect present indicators would potentially lead to better inference of future behavior tendencies)
- **The fraction of actually executed missions for which a reliable measure of performance exists** (for how many missions do we have the metrics needed to assess mission performance? for any given mission, how many salient performance metrics are we actually capturing for that mission?)
- **The number of lessons-learned distributed** (clearly depends on how lessons and distribution are counted)

4) Meta-Metric Learning

Meta-metrics describing how well the relevance, coherence and planning assessment metrics support plan assessment, and enable improvement of the metrics:

- **Inter-phase correlation** (e.g., correlation of lessons-learned distributed and follow-on planning preparation metrics over time might suggest little relationship between the two. Perhaps the value of the lesson should be included in the metric, or independently, whether the lesson-learned changed any process)
- **Intra-phase correlation** (e.g., no correlation between asset attrition and assessment of satisfaction of goal state suggests that it might valuable to distinguish between "productive" and "unproductive" attrition)

VI. Ontology-driven Querying of Plan information

The kinds of representations described above are necessary to support Living Plan requirements. But they are not sufficient. Without the query support to populate them, the representations are vacuous. Since the underlying living plan-related data requires the inference-based identification of objects with associated attribute and location information under conditions of uncertainty, ontology-driven query mechanisms will need to include probabilistic functions in addition to more traditional deductive ones.

Consider the following metric where we have underlined ontology terms to be used by the Living Plan framework:

> the percentage of operational anti-aircraft missile sites by area-of-operations for some given plan specification.

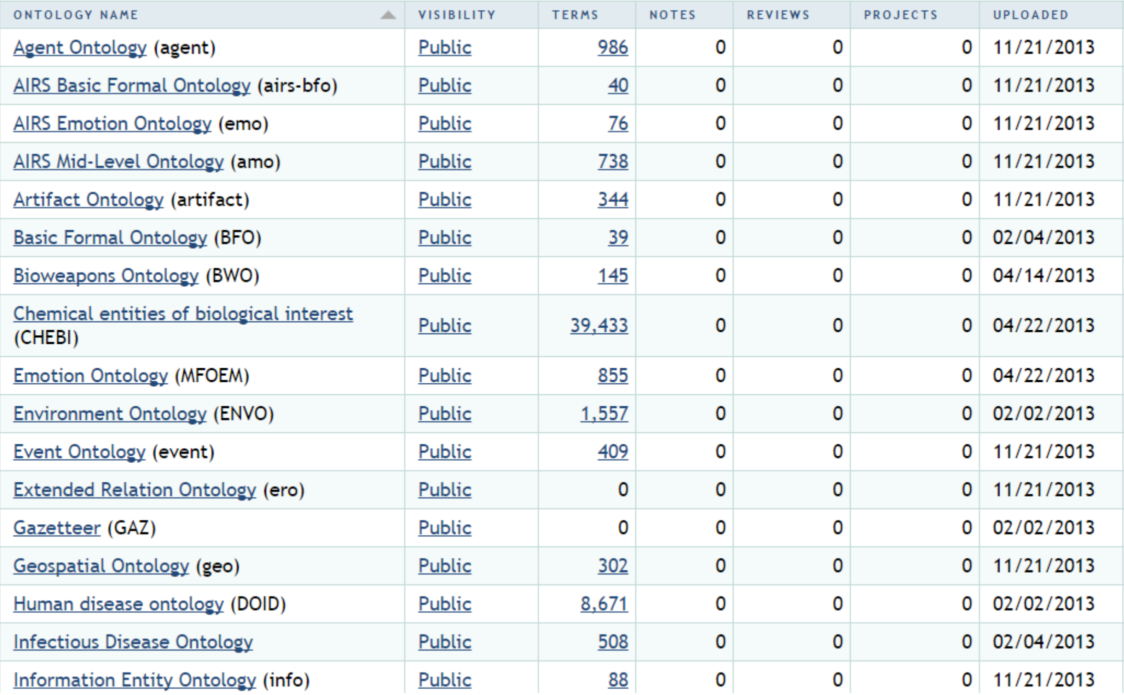
Such a metric would be useful in determining the progress of an operational objective for example related to suppression of air defenses. Though seemingly straightforward, even this metric raises a number of interesting semantic challenges that need to be resolved by a query processor.

As stated, the metric is conditioned on a user's specification of a plan. Given a plan, the metric represents the percentage of operational anti-aircraft missile sites by area-of-operations for the specified plan. The query processor thus needs to be able to ascertain area-of-operations associated with a given plan, something which could possibly vary over time.

A. *Indirect identification of plans*

Even the identification of the plan may be a non-trivial exercise. While in theory it may be possible to use a unique plan identifier to locate the desired plan, in practice the plan may be identified indirectly in a number of ways, such as:

- Attributes: Using combinations of attributes such as *plan phase* (development, execution or post-execution), Commander in charge of plan execution, approval date, and so on.
- Containment: Identifying related plans through relations of *containing* or *being contained within* other plans: the AOP (Air Operations Plan) is *contained within* a specified Joint Campaign Plan, or conversely, for a Campaign Plan that contains a specified AOP.
- Assets: By relating a plan to the assets associated with it during a given time frame, as when an AOP is tasking Squadron X in some given week.
- Operational relation: For example, one plan *precedes* or *succeeds* another as *pre-condition* or *sequel*. Or two plans relate to each by having mutually dependent executions.



ONTOLOGY NAME	VISIBILITY	TERMS	NOTES	REVIEWS	PROJECTS	UPLOADED
Agent Ontology (agent)	Public	986	0	0	0	11/21/2013
AIRS Basic Formal Ontology (airs-bfo)	Public	40	0	0	0	11/21/2013
AIRS Emotion Ontology (emo)	Public	76	0	0	0	11/21/2013
AIRS Mid-Level Ontology (amo)	Public	738	0	0	0	11/21/2013
Artifact Ontology (artifact)	Public	344	0	0	0	11/21/2013
Basic Formal Ontology (BFO)	Public	39	0	0	0	02/04/2013
Bioweapons Ontology (BWO)	Public	145	0	0	0	04/14/2013
Chemical entities of biological interest (CHEBI)	Public	39,433	0	0	0	04/22/2013
Emotion Ontology (MFOEM)	Public	855	0	0	0	04/22/2013
Environment Ontology (ENVO)	Public	1,557	0	0	0	02/02/2013
Event Ontology (event)	Public	409	0	0	0	11/21/2013
Extended Relation Ontology (ero)	Public	0	0	0	0	11/21/2013
Gazetteer (GAZ)	Public	0	0	0	0	02/02/2013
Geospatial Ontology (geo)	Public	302	0	0	0	11/21/2013
Human disease ontology (DOID)	Public	8,671	0	0	0	02/02/2013
Infectious Disease Ontology	Public	508	0	0	0	02/04/2013
Information Entity Ontology (info)	Public	88	0	0	0	11/21/2013

Figure 2: I2WD ontologies at http://milportal.org

One or more of these methods could be used in the query to identify the desired plan, requiring the query processor to apply additional knowledge of plan attributes and relations to properly parse the query to eventually locate the desired plan and its area-of-operations.

B. Ontology-driven queries

The complexity and dynamic nature of relationships between the plans and the involved information cannot be adequately represented in non-semantic technologies (for example in traditional databases). Moreover, direct traditional querying of such representations will be difficult to automate and maintain in the necessary flexible manner, and the results of such querying may not be capable of the needed rapid update to incorporate emerging important data. Our hypothesis, therefore, which draws on the work described in [7,8] is that a comprehensive and incrementally evolving set of Living Plan ontologies, drawing on the I2WD suite of ontologies (see Figure 2) can provide the needed nuanced representation of the plans, metrics, and of the semantics of the source data against which the querying is performed, while taking account of relationships between all of these components. Such an approach will lay a foundation for sophisticated querying and analytics enhanced by inference, and is designed above all, to enable agile changes to all components. Additionally, the ontology framework will have to include representations of complicating factors such as those described below and their relationships with the plans and metrics.

C. Probabilistic ontological classifications

One example complication concerns the identification of the location constraint for those sites that are to be considered because they lie within the area-of-operations. The problem turns on the fact that there may be sites physically outside this area that are identified as harboring capabilities that project into the area-of-operations. This may imply an ambiguity at the operational level. If the focus is on assessing the performance of missions to disrupt or destroy sites physically within the area-of-operations, then the metric should be interpreted in one way. If, on the other hand, the intent of the metric is to assess the security of aircraft within the area-of-operations, then the better interpretation may extend the focus to include sites that have an air defense capability that reaches into the area-of-operations from outside. In order to properly respond to a query based on the latter interpretation, the system would need to be able to infer such projection capabilities and perform spatial reasoning to find substantial intersections with the physical boundaries of the area-of-operations. Such capabilities may depend on the type of missiles available, requiring further information about specific missile capabilities and deployment.

Another potential complication is bias in the identification of individual sites for counting. For example, the adversary might expend additional effort to hide remaining operational sites rather than sites that may have already been degraded in some way. Conversely, missile firings from operational sites make them more difficult to hide. At the same time, own-forces may not expend as much effort in identification and counting of non-operational sites as those which still pose a threat. In short, the process of identifying and counting sites may be substantially different according to whether they are operational or non-operational. To provide appropriate measures of confidence in the associated metrics, the query processor would have to know what sorts of biases to consider and their relative magnitudes in terms of attributes such as power projection capability, which will be defined in our ontology framework.

A likely more difficult counting complication would arise from semantic assembly of information regarding the very attribute of being *operational* as applied to sites. Whether a site is operational may be difficult to determine for multiple reasons. For example, if a site loses some part of its targeting capacity but retains ability to launch, then it is operational as a launch site, but without targeting it will pose little threat to modern aircraft. The state of the site may also be time-dependent; for example, a site that is partially degraded could be anticipated to be restored at some point in the future. Such expectations would depend on the nature of the degradation and the resources available to make repairs and restore operation. At any particular time, the query processor would have to combine operational state attributes based on reports from different times and with varying levels of confidence arising from uncertainty in expectations as to whether a site will remain operational.

Other complications might arise in classifying a site as functioning or not functioning as an ‘anti-aircraft missile site’. It is certainly possible that the raw intelligence information and sensor data on which counts are made will not directly and unambiguously classify a facility as an anti-aircraft missile site. Instead, there may be reports of a more specific nature (for example, that we are dealing with a specific type of missile capability) which through interaction with weapons ontology would be determined to qualify more generally as ‘anti-aircraft’. On the other hand, some reports may refer only to a ‘missile site’, which would then require further inference to determine if the site is likely to have a more specific type of anti-aircraft capability. Such inferences generally require the knowledge of type-subtype relations and the attributes on which such classifications are based. For example, information about a missile site supertype could be inferred to be also of the anti-aircraft missile site subtype through examination of other potentially known attributes, such as size and location of the site, imagery features, connectivity to other assets, and so on. Such information will be incorporated as probabilistic functions into our ontology framework.

D. Missing, inconsistent and other invalid data

Considering the fog of war, some information will at any given stage be incorrect, inconsistent, or missing. Barring independent evidence to the contrary, incorrect information, such as a site being reported as operational that is not, cannot be rectified. However, when there are multiple reports in conflict, it may be possible to reach a most likely conclusion. A query processor that maintains, or has access to, meta-information regarding the typical or historically-observed believability of reports from various sources can combine conflicting reports as weighted evidence to reach a most believable conclusion. The needed provenance-related attributes, too, will be incorporated into our ontology framework.

A conflict in evidence may be due to understandable reasons, the simplest being that they were made at different times in relation to something that is changing, such as the state of a missile site. A more complex case would involve the ability of different sources to provide substantial evidence at different times or under different circumstances. For example, prior to actually observing an anti-aircraft missile site launch a weapon, a determination of its state of operation may be difficult to establish. An intact-looking site might be non-operational for reasons that are not directly observable, such as broken electronic or computer-based equipment. Under these circumstances, direct observation might provide credible evidence of non-operational status (the physical structure may be visibly degraded or destroyed), without being able to provide evidence of operational status. Intelligence reports from intercepted communications would be a better source of information under these circumstances, but only if they are to be believed as genuine and not intentional misinformation. Of course, direct observation of a successful missile launch at a later point in time would over-rule any prior assertions about the site's state, but only until contravening reports are later received indicating that its state may have changed, such as a battle damage assessment that it was successfully struck and destroyed at an even later point.

Such issues, related to reports of the changing state of a missile site, may be interpreted differently depending on the purpose of the associated metric. If the intent is to assess progress of given actions toward an operational objective of reducing the risk of operations in a given airspace, then the most important information is the conversion through those actions of known operational sites into non-operational sites. In that case, for example, it would be less important to know which sites were non-operational for other reasons prior to the start of the campaign. At the same time, the change in state of a particular site would presumably be the effect of some action taken, and such information would aid in the interpretation of the action reports. For example, if the site were observed to be launching missiles prior to a kinetic strike on the facility and no launches were observed after the strike, it would be reasonable to believe that the strike had its intended effect in rendering the site non-operational. On the other hand, if the metric is being used primarily to ascertain the relative risk of operations in that airspace, then the numbers of operational and non-operational sites prior to the campaign become important, as well as the previously-discussed issue of sites being restored to operation over time.

In addition to incorrect and conflicting information, the query processor must also deal with missing information. In some circumstances reports may be available only for certain time periods, or concerning certain types of information. For example we may have reports on site location without state of operation information, or only assertions of being operational but not of being non-operational. Such differences in missing information will add complexity to making a reliable estimate of the ratio of operational to non-operational sites over a given area of interest.

VII. Conclusion

To support the Secretary of Defense's vision for Living Plans, we believe that plan-related ontologies need to be extended into two areas:

- A generic planning process ontology that is based on the Information Artifact Ontology and that takes into account the cyclical process of planning.
- Ontologies containing representations of each of the kinds of attributes and relations needed to identify desired plans according to relevant areas-of-operations, assets, capabilities, and so forth.

Additionally, the query processing component of any plan-related computational framework that converts potentially huge stores of plan-related expressions (data types, values, natural language expressions), into user-oriented actionable metrics needs to be aware not merely of the ontologies, but also of the needed types of deductive transformations and, as we showed above, of probabilistic classifications. Materialized query processing tools will rely on the principles set forth in [7, 8] which are being used to integrate diverse data in a variety of disciplines. The approach is designed to achieve integration in an agile, flexible and incremental way, and also to incorporate into our system the ontology content created for related purposes by our collaborators in different military domains and disciplines.

References

[1] Mintzberg, H. (1994). The fall and rise of strategic planning. *Harvard Business Review*, 72, 107-114.

[2] Grant, R. M. (2003). Strategic planning in a turbulent environment: evidence from the oil majors. *Strategic Management Journal*, 24, 491-517.

[3] Davis, P. K. (2012). Lessons from RAND's Work on Planning Under Uncertainty for National Security. DTIC Document.

[4] Boukhtouta, A., Bedrouni, J., Berger, J., Bouak, F., and Gulati, S. (2004). *A Survey of Military Planning Systems. International Command and Control Research and Technology Symposiu*m (ICCRTS).

[5] Pollack, M. E. and McCarthy, C. (1999). Towards focused plan monitoring: A technique and an application to mobile robots, *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation* (CIRA99), 144-149.

[6] Boutilier, C. (2000). Approximately optimal monitoring of plan preconditions. *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, 54-62.

[7] Smith, B., Malyuta, T., Mandric,k W.S., Fu, C., Parent, K., Patel, M. (2012). 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 (CEUR 996), 112-119.

[8] Salmen, D., Malyuta, T., Hansen, A., Cronen, S., Smith B. (2011). Integration of Intelligence Data through Semantic Enhancement, *Proceedings of the Conference on Semantic Technology in Intelligence, Defense and Security* (STIDS), George Mason University, Fairfax, VA, November 16-17 (CEUR 808), 6-13.

Effective RDF Resource Identifiers for Integration of Structured Data Sources

Ian Emmons
Raytheon BBN Technologies
Arlington, VA
iemmons@bbn.com

Abstract—**Based upon extensive experience in the use of semantic technologies to integrate structured data from disparate systems, the author recommends a set of best practices for creating IRIs for RDF resources. Particular attention is paid to avoiding unnecessary coreferences in scenarios where data is drawn from a structured, non-semantic source of record, issues that commonly arise in Department of Defense (DoD), Intelligence Community (IC), and government contracting scenarios, as well as other common pitfalls.**

I. Background

At the very foundation of the Resource Description Framework (RDF), before we can ever write down a single triple, we encounter the notion of the International Resource Identifier (IRI) as a means to create identifiers with global validity.[1] The Web has proven that the IRI is in fact a good solution to this problem, and so the RDF standard has very little further to say about this topic [3]. However, creating sub-optimal IRIs is a common pitfall of the Semantic Web.

In the discussion that follows, an important concept is the *source of record* of a datum. This is the particular copy of the datum that is considered to be its authoritative source. When data is created directly in RDF format, so that the source of record is the RDF itself, then choosing good IRIs is relatively straightforward. However, the source of record for a data set is often a non-RDF database of some sort, such as a relational database, and the data is translated into RDF in order to enable Semantic Web processing techniques. In these cases, there are a number of additional considerations that come into play when choosing resource IRIs, which we address here. Note that a different set of issues arises when choosing IRIs for data whose source of record is unstructured. Such situations are not considered here.

In the sections below, we will first address the issues associated with forming IRIs in general, and then we will consider structured, non-RDF sources of record. We pay particular attention to situations that arise in DoD, IC, and government contracting.

II. Considerations for Any Source of Record

This section discusses issues that apply to data from any kind of source of record.

A. Hierarchical Naming

The most important aspect of resource identifiers is this: A resource's IRI must be globally unique. In other words, no two resources may share the same identifier. Unlike the primary key of a database table, which need only be unique among the records of that table, a resource's IRI must be globally unique. There are two widely used systems for creating such identifiers: Globally Unique Identifiers (GUIDs), which are sometimes also called Universally Unique Identifiers (UUIDs), and hierarchical naming. The former gathers a number of relatively unique items from the local computing environment, such as the current time and network interface MAC addresses, and combines them algorithmically into a large, fixed-length number whose uniqueness is guaranteed with such high probability that we can assume absolute global uniqueness.

RDF uses hierarchical naming to achieve global uniqueness, as exemplified by the IRI system. This approach constructs an identifier as a variable-length character string consisting of a hierarchy of segments, each of which further narrows the scope until a unique identifier for a specific item is achieved. Each successive level of hierarchy carves out a subset of the namespace denoted by its predecessor and often corresponds to an organizational entity with jurisdiction over that subset. For instance, the segments might be arrayed as follows:

```
http://org/dept/project/class/item
```

The portion preceding the first colon designates the scheme. Most RDF IRIs use the `http:` scheme as shown here, but others are possible as discussed below in Section II-B. The "org" portion is a Domain Name System (DNS) name (see Section II-C). Using a DNS name leverages the domain registration process to reserve a namespace on behalf of an organization. From there, the IRI narrows the scope by appending a department name, a project name, the name of a class of entities, and then finally the identifier of one item within that class.

Naturally, there are many variations on this theme:

- In a very large organization, the "dept" segment may be replaced by several segments that descend through multiple layers of organizational structure. And some organizations may prefer to use a sub-domain of their primary DNS name for this purpose.

[1] The IRI [1] is a generalization of the Uniform Resource Identifier (URI) [2] to include international character sets.

- The "project" portion of the IRI may also be subdivided into components.
- Including a date can help with versioning, and it can guard against the possibility that the remainder of the identifier is reproduced at a later time by a different organization that has the same name.
- Many practitioners include a segment immediately after the domain name that is either "id" or "ontology" to distinguish between instance data and its ontology.

There are numerous sources that offer guidelines for constructing such hierarchical names [4, 5, 6, 7], which the reader may wish to consult to gain a deeper understanding of best practices. However, there is no one-size-fits-all strategy, so you will need to adapt the given advice to your particular situation.

As a point of terminology, the portion of the IRI in the example above that precedes "item", including the last slash, is called the *base IRI*. The base may end in a slash, as in this case, or with a hash '#'. For our purposes here, the distinction is immaterial, but a full discussion of the differences can be found in [4].

In order for hierarchical naming to properly achieve its goal of global uniqueness, it is crucial that each RDF author create new IRIs only within those hierarchical scopes in which he or she has the authority to do so. For instance, an author who works for Company A should not create IRIs using the domain name of Company B, unless Company B has given its permission to do so. Otherwise, there exists a very real possibility for different authors to use the same IRI to identify two different things. Likewise, an author within one department of a company should not create IRIs using the department identifier segment of a different department, unless the second department has given its permission to do so.

A different situation arises when one RDF author has already created an IRI for an entity and another author wants an identifier for the same item. In this case, the second author should, whenever possible, reuse the original author's IRI in order to avoid the confusion that arises from having multiple names for the same entity. (See Section III-A below.)

Note that although the RDF standard chose IRIs as its system for unique identification, RDF authors can still use GUIDs, and Section II-B shows how.

B. IRI Schemes

In Section III-A above, it was noted that the IRI scheme most commonly used in RDF is `http:`. While this is true, there are other schemes that work well. But first, why are `http:` IRIs so common? In part the answer is that this scheme possesses a mix of features that make unique identification easy, with extremely low cost. In addition, `http:` IRIs can be resolvable, which means that given appropriately configured infrastructure, the resource identifier can also be used to retrieve information about the resource it identifies. A detailed discussion of how to achieve this goal can be found in [4].

Another IRI scheme that is useful in RDF is the "tag" scheme. The syntax of `tag:` IRIs is given in [8], and an accessible discussion of how to create them can be found in [9]. The `tag:` and `http:` schemes are similar in many ways. The principle ways in which they differ are the following:

- `tag:` IRIs are explicitly non-resolvable.
- The root of a `tag:` IRI may be a domain name, as with `http:`, or an email address.
- The `tag:` scheme formalizes the use of dates in the IRI.

Our original IRI example, translated to the `tag:` scheme, looks like so:

```
tag:org,2014-10-01:dept/project/class/item
```

where "org" is either a DNS name as before or an email address.

IRI schemes other than `http:` and `tag:` are very rare in RDF. One other scheme that might seem useful is the Uniform Resource Name (URN) scheme [10]. However, URN IRIs require the registration of a namespace [11], which makes them far too cumbersome for use in RDF. However, there is a URN namespace already declared for GUIDs [12]. Using this, RDF authors can easily convert a GUID into a valid IRI simply by prepending the string `urn:uuid:`, like so:

```
urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6
```

Of course, a GUID can also be converted into a valid IRI by prepending an `http:` or `tag:` base IRI.

C. DNS Names

This section discusses a number of issues involving the DNS name portion of an IRI so that developers can better avoid them.

Example Domains: Occasionally, an IRI author may not have a DNS name to which he or she can lay claim. Unless the work is truly intended as an example, it is best to avoid using the reserved names `example.org` or `example.com`. IRIs have a tendency to rapidly work their way into many nooks and crannies of a software system (such as queries, source code, or rules), so that the global search-and-replace operation to switch away from `example.org` that seems so easy at the beginning quickly becomes a large undertaking that inconveniences the whole development team. Instead, either acquire a proper domain name or use `tag:` IRIs containing an email address.

Contractors: Contractors who are working on behalf of another organization may be required to use the DNS name of their customer. In such cases, contractors should take care to be sure that the appropriate individuals in the customer organization know that this is happening so that naming collisions do not occur. One easy and effective way that the customer organization can solve such problems is to reserve to the contractor a subset of their IRI space by allocating an IRI segment beyond the DNS name to the contractor's project.

Resolvable IRIs: A popular way to expose semantically encoded data is via the Linked Open Data (LOD) methodology. One of its tenets is that RDF IRIs should always be resolvable, and that when resolved, an IRI should return some information about the concept it identifies. A discussion of best practices for implementing such systems can be found in [4]. A number of issues can crop up in these cases that developers should be careful to consider:

- Resolvable IRIs require a DNS name that resolves to an actual Web server. In a large enterprise such as the DoD or IC, these tasks may require considerable administrative effort and entail non-trivial approval periods. Some cases may also require security authorizations. Thus, developers must establish DNS names and servers as early as possible in order to avoid changing IRIs mid-project.
- Be sure to check early in your project whether organizational security policies require that web traffic runs over secure connections. This is a frequent requirement on DoD and IC projects. In such cases, your IRIs will need to use the `https:` scheme.
- DoD and IC projects that result in an operational system typically must establish development and test deployments in addition to the operational system itself. These will necessarily have different DNS names, resulting in separate IRI spaces. Queries, source code, and rules, however, will by default contain a single DNS name, and will therefore break when moved from development to test unless a more sophisticated approach is devised. One partial solution is to segregate the ontologies on a separate server that is common to all of the deployed systems. Since the ontologies are typically read-only static content, this server can be an incredibly simple deployment that never changes. Since many of the IRIs that appear in queries, source code, and rules are taken from the ontologies, this will substantially decrease the number of IRIs that break when moving between deployments.
- Classified systems must sometimes run on multiple, isolated networks. However, the entities that are identified on the lower networks are typically also present on the upper ones. Thus, the IRIs that identify these entities should be the same in all cases. In an LOD scenario, this means that corresponding servers on different networks should have the same DNS name.

D. Allowed Characters

One confusing point about RDF IRIs is that the portion after the base IRI is allowed to contain a broad range of characters directly and nearly any character in an escaped (or percent-encoded) form. However, as a practical matter, the characters allowed in this portion of the IRI are limited to the following:

- The first character must be a letter or underscore, and
- Subsequent characters must be letters, underscores, hyphens, periods, or digits.

The reason for this is that in many contexts the base IRI is assigned a short *prefix,* which is used to abbreviate. For instance,

```
http://org/dept/project/class/item
```

can be written more concisely as `p:item` in a document that declares a prefix like so:

```
prefix p: <http://org/dept/project/class/>
```

This prefix syntax is borrowed from eXtensible Markup Language (XML) namespaces, where the abbreviated form is called a *QName* (qualified name), and the portion following the prefix is called a *local name.* The XML syntax for a local name is much more restrictive than the IRI syntax, as noted above. Thus, for reasons of convenience, most RDF authors limit themselves to the more restrictive syntax so that they can use prefixes to abbreviate their IRIs. This is particularly helpful for making SPARQL Protocol and RDF Query Language (SPARQL)[2] queries readable [13], but it also helps when data is viewed in Turtle (`.ttl`) syntax [14].

E. IRIs as Content

While forming IRIs, it is natural and proper to put some thought into their information content. However, at query time, IRIs should be considered to be opaque identifiers devoid of content other than their ability to uniquely identify a resource. The reason for this is that semantic query languages and query processors are poorly adapted to efficiently parsing out pieces of IRIs. Therefore, if there is information embedded in your IRIs that you wish to access from queries, duplicate that information in properties designed for that purpose.

III. IRIs for Non-RDF Sources of Record

Over the last decade, Semantic Web technologies have proven to be an effective approach for solving data integration problems. Usually this involves creating an RDF representation of data from an existing, non-semantic source of record. The RDF may then be stored in, for example, a triple store such as Raytheon BBN Technologies's (BBN's) Parliament™ [15],[3] or it may be generated on demand and fed dynamically to the requesting activity, as with a federated query system like BBN's Asio™ [16].

Whether stored or generated on demand, the end result is that the same data is represented in multiple formats and places, and in such situations the proper creation and maintenance of identifiers is crucial. Our goal in such cases is always to maintain a one-to-one relationship between entities in the source of record (which may be any kind of identifier) and their identifiers in other representations (which are always IRIs in our case). In this section, we consider how best to accomplish this goal in general, as well as investigate a number of specific examples.

A. Uniqueness and Reproducibility

To achieve the goal given above, we put forth two guiding principles for the formation of IRIs:

- *Unique:* A resource's IRI must be globally unique, as discussed above in Section II-A.
- *Reproducible:* Every time we form the RDF representation of an entity, the IRI we create must be same one.

[2] The SPARQL acronym was chosen during a time when recursive acronyms were in vogue. As a result, the expansion of the acronym is confusing because it contains the acronym itself.

[3] http://parliament.semwebcentral.org/

These two tenets correspond to the two halves of the one-to-one constraint that is our goal. The uniqueness principle says that no two data items may share the same identifier, and the reproducibility principle says that no data item may have two distinct identifiers.

Most people understand the uniqueness principle easily, both the need for it and how to achieve it. The reproducibility principle, however, is more subtle: If we encounter the same entity twice, forming its RDF data representation each time, then we must form the same IRI for it both times. But why is this important?

Consider an example system that queries a Relational Database Management System (RDBMS) and then translates the result set into RDF. A straightforward way to do this is to create an IRI to represent each row (or, more precisely, to represent the real-world entity represented by the row), and then transform each column into a property of that resource. Foreign key columns become object properties, and other columns translate into datatype properties. So if we issue the following query:

```
select employee_id, first_name, last_name,
  ssn from Employee where employee_id < 40
```

and the result set looks like Table I, then the resulting RDF might look like so:

employee_id	first_name	last_name	ssn
12	Robert	Smith	123-45-6789
37	Alice	Jones	987-65-4321

TABLE I
EXAMPLE RESULT SET

```
prefix ont: <http://example.org/ont#>
prefix id: <http://example.org/id#>
id:z138ce39f-0434-4d16-b307-82b9206142b5
  a ont:Employee ;
  ont:employeeId 12 ;
  ont:firstName "Robert" ;
  ont:lastName "Smith" ;
  ont:ssn "123-45-6789" .
id:z1e036a52-7e1e-4a33-a48f-03837634f776
  a ont:Employee ;
  ont:employeeId 37 ;
  ont:firstName "Alice" ;
  ont:lastName "Jones" ;
  ont:ssn "987-65-4321" .
```

Note that the IRIs for the two employee instances are based on GUIDs.

Now, suppose that later on in the execution of this same system, another query against the Employee table happens to return row 37 again. Then the same code will translate the new result set into RDF. When it returns to row 37, it will get a GUID just as it always does, generating something like this:

```
id:zf4139560-8c48-4b4c-a860-5d1bb9e02bdf
  a ont:Employee ;
  ont:employeeId 37 ;
  ont:firstName "Alice" ;
  ont:lastName "Jones" ;
  ont:ssn "987-65-4321" .
```

This is exactly the same RDF as before, except that by the nature of GUIDs, a different IRI is now representing employee 37. The end result is that in RDF, we now have two separate employees named Alice Jones with employee number 37. In other words, we have created a coreference where none existed before. This is exactly what the reproducibility principle seeks to avoid — unnecessary coreferences — and it also illustrates why GUIDs are best avoided in RDF resources under most circumstances. (For an exception, see Section III-B.)

One way to avoid the coreference created in this scenario is to identify employee 37 by the IRI `id:employee37`. Using the primary key from the RDBMS table ensures that every time we encounter row 37, we will form the same IRI. But what if we encounter Alice Jones in a different context, say in an RDBMS table at the Internal Revenue Service (IRS)? Now we cannot expect that Alice will be associated with the number 37, because that number is an internal implementation detail of her employer's database. In order to avoid creating a coreference in this case, we might turn to Alice's Social Security Number (SSN). This information will almost certainly be in both databases, because both the IRS and Alice's employer are concerned about her income tax. Thus if we identify Alice by the IRI `id:ssn-987-65-4321`, we can be sure that Alice will be a single, unified entity across these two separate organizations.

Generalizing from this particular example, we see that to comply with the reproducibility principle, the information used to form a resource's IRI should be semantically intrinsic to the thing being identified. Ideally, this should hold true whether we encounter the entity in a repeat of the original context (e.g., the same source of record) or in a different context altogether (such as a source of record in a different organization).

Clearly, if we encounter the entity in two *very* different contexts, we may discover that there is no identifying information held in common. The scenario above was cleverly constructed so that a solution satisfying the reproducibility principle was readily available. However, if we add a third database to this scenario from the immigration agency of a non-U.S. government, then the record of Alice's visit while on vacation will almost certainly not contain her SSN.

Thus, it is important to realize that while the uniqueness principle is of paramount importance, never to be violated, the reproducibility principle is really much more of a guideline to strive for, but which usually requires some carefully chosen compromises.

B. Example Scenarios

In this section, we seek to explore a variety of data integration scenarios to see how the uniqueness and reproducibility principles can best be achieved.

Derivation from a single source of record: In cases where the RDF will always be derived from a consistent source of record, we look to the identification system used in the source of record itself. One common case is a database table that uses a sequence number for its primary key. In this case, we can use the primary key itself to form the resource identifier, but as with the example given in Section III-A, this will cause a coreference whenever the same real-world entity is encountered in a different source of record. Thus, if the table contains a semantic key, i.e., a key that has semantic meaning intrinsic to the entity represented by the table, that may be preferable.

Note that a key, and particularly a semantic key, can span several columns. In such cases, the individual values from these columns must be combined to form the IRI. Section III-C shows one way to do this correctly and without producing long and unwieldy IRIs.

Occasionally, you may find a database table whose primary key is a GUID column. (This happens most often with Microsoft SQL Server.) This is one occasion when using a GUID in your IRIs is a reasonable thing to do, because the GUIDs come from the source of record, rather than being generated anew every time the IRI is created.

Derivation from multiple sources of record: Sometimes there are multiple sources of record that contain overlapping data sets. If the sources of record were built within the same organization, they may have a common identifier system that can be leveraged to create consistent IRIs across the sources of record. Alternately, the entities in question may have a well-known standard system of identification, such as airplane tail numbers or merchant ship registration, that is included in all of the sources of record. However, in the general case of multiple sources of record that were built independently, the sources of record will not have a common system of unique identifiers. In such cases, it may be necessary to encode data from each source of record in isolation, resulting in potential coreferences, and then apply a coreference resolution algorithm to identify and merge the coreferences after the fact. Such algorithms are beyond the scope of this paper.

Flat file sources of record: When a flat file is nicely designed, it presents no issues that are not covered by the scenarios discussed above. Unfortunately, flat files are often more ad hoc than databases, with little thought given by their creators to identification of the entities contained within. As a result, it is common when translating a flat file into RDF to discover that there is no column that serves the purpose of identifying the row.

If possible, try to identify a subset of the columns in the file that uniquely identify the entity represented by the row. In particularly difficult cases, the author has resorted to regarding all of the columns as the key. This approach will tend to create coreferences, but it has the best chance of satisfying the uniqueness principle.

An approach that is not recommended is to use the flat file's name and/or path to form part of the IRI for each row. This is usually not a good idea because the file name and location can be changed without any change to the file content, and therefore without a change to the semantics of the entities represented therein. In other words, every time the file is moved or renamed, there is potential to create coreferences for all of the contained entities.

Derivation from a single source of record, with intermediate processing: Consider a case where there is a single source of record, like a database, that feeds data through some process (or set of processes), transforming or enriching the data on the way, and then we wish to render the final output as RDF. By far the best approach to forming IRIs in such a case is to find the identifying information from the original source of record and make sure that this information is carried throughout the processing chains. This enables the IRIs to be constructed independent of the particular processing steps, and it also allows a consistent IRI for an entity that passes through multiple processing chains, thereby avoiding unnecessary coreferences.

Sub-row entities: In simple cases, each row in a database table translates into one RDF entity, as outlined in Section III-A. However, ontologies are often more structured than database schemas, and so what appears as just more columns in a database table may well be a separate entity in your ontology. Thus, it is often the case that one database row translates into multiple related resources, each with its own properties, in RDF.

Some sub-row entities are logically part of the entity represented by the row in which they occur. In such cases, a handy way to form the IRI is to use the IRI for the row entity and then append additional key fields to distinguish it from the row entity.

Other sub-row entities are logically independent of the entity represented by the row in which they occur. An easy way to distinguish whether a sub-row entity falls into this category is to ask yourself the following question: "If the columns containing the sub-row entity in two rows contain the same values, should the end result be two row entities related to one sub-row entity?" If the answer is yes, then the sub-row entity is logically independent. In this case, you will want to form the IRI for the sub-row entity from only those columns containing the sub-row entity much as if they were a row in a separate table.

C. Avoiding Overly Long IRIs

As indicated in the above example scenarios, sometimes many individual pieces of information must be combined into a single IRI. This can result in an enormously long IRI, and it can also result in arbitrary characters that must be escaped. A handy way to avoid this situation is to concatenate the individual strings and then run the result through a cryptographic message digest (or hash) algorithm. Sample Java code for this procedure is shown in Figure 1.

There are a couple of subtleties to this procedure that require some explanation. First, the use of a hash weakens the uniqueness guarantee. However, cryptographic hash algorithms are designed specifically to avoid collisions, and so the probability of this procedure causing a collision of IRIs is vanishingly small when a strong algorithm such as SHA-256 is used.

Second, a quick examination of the code in Figure 1 reveals that the individual pieces of key material are not simply concatenated, but rather encoded as if they were a row within a Comma-Separated Values (CSV) file. The reason for this is that string concatenation is not an invertible operation. For instance, if we concatenate “their reversible”, we get exactly the same result as if we concatenate “the irreversible”. However, when we CSV-encode these two pairs of strings, we get “their,reversible” and “the,irreversible”, which are distinct. Thus, the use of CSV encoding upholds the uniqueness principle by assuring that two distinct keys are not mapped to a single IRI.

```
static String encode(List<String> keys)
{
  StringBuilder buf = new StringBuilder();
  boolean isFirstKey = true;
  for (String key : keys)
  {
    if (!isFirstKey)
    {
      buf.append(’,’);
    }
    if (key.contains("\""))
    {
      buf.append(’\"’);
      buf.append(
        key.replace("\"", "\"\""));
      buf.append(’\"’);
    }
    else if (key.contains(",")
      || key.contains("\r")
      || key.contains("\n"))
    {
      buf.append(’\"’);
      buf.append(key);
      buf.append(’\"’);
    }
    else
    {
      buf.append(key);
    }
    isFirstKey = false;
  }
  try
  {
    MessageDigest md = MessageDigest
      .getInstance("SHA-256");
    byte[] input = buf.toString()
      .getBytes("UTF-8");
    HexBinaryAdapter hba
      = new HexBinaryAdapter();
    return hba.marshal(md.digest(input));
  }
  catch (NoSuchAlgorithmException
    | UnsupportedEncodingException e)
  {
    // This should never happen, because
    // all JVMs must support the SHA-256
    // hash and UTF-8 char encoding.
    throw new RuntimeException(e);
  }
}
```

Fig. 1. Hashing a List of Strings for Inclusion in an IRI

IV. Conclusion

Though the IRI lies at the heart of the RDF standard, creating IRIs for RDF resources is a topic that is often glossed over in the literature. In this treatment, we hope to have given the reader a solid understanding of the issues underlying the creation of effective IRIs, as well as specific advice for a range of scenarios relating to structured, non-RDF sources of record as well as situations that arise in DoD, IC, and government contracting.

V. Glossary

Asio	Asio$^{\text{TM}}$ is BBN’s semantic federated query framework. This is not an acronym. It is simply a name derived from a genus of owls. (p. 3)
BBN	Raytheon BBN Technologies, Inc. (p. 3)
CSV	Comma-Separated Values (p. 6)
DNS	Domain Name System (pp. 1–3)
DoD	Department of Defense (pp. 1, 3, 6)
GUID	Globally Unique Identifier (pp. 1, 2, 4, 5)
IC	Intelligence Community (pp. 1, 3, 6)
IRI	International Resource Identifier (pp. 1–6)
IRS	Internal Revenue Service (p. 4)
LOD	Linked Open Data (pp. 2, 3)
Parliament	Parliament$^{\text{TM}}$ is BBN’s triple store, so named because “parliament” is the collective noun for a group of owls. A triple store is a specialized database tuned to the unique needs of the Semantic Web data representation. (p. 3)
RDBMS	Relational Database Management System (p. 4)
RDF	Resource Description Framework (pp. 1–6)
SPARQL	SPARQL Protocol and RDF Query Language. This acronym is a bit confusing, because it was conceived when recursive acronyms were popular. (p. 3)
SSN	Social Security Number (p. 4)
URI	Uniform Resource Identifier (p. 1)
URN	Uniform Resource Name (p. 2)
UUID	Universally Unique Identifier (p. 1)
XML	eXtensible Markup Language (p. 3)

VI. References

[1] M. Düerst and M. Suignard, “Internationalized Resource Identifiers (IRIs),” IETF, Request for Comments 3987, Jan. 2005. [Online]. Available: http://tools.ietf.org/html/rfc3987 (cit. on p. 1).

[2] T. Berners-Lee, R. T. Fielding, and L. Masinter, “Uniform Resource Identifier (URI): Generic Syntax,” IETF, Request for Comments 3986, Jan. 2005. [Online]. Available: http://tools.ietf.org/html/rfc3986 (cit. on p. 1).

[3] G. Schreiber, Y. Raimond, F. Manola, E. Miller, and B. McBride, “RDF 1.1 Primer,” W3C, Working Group Note, version 1.1, Jun. 24, 2014. [Online]. Available: http://www.w3.org/TR/rdf-primer/ (cit. on p. 1).

[4] L. Sauermann, R. Cyganiak, D. Ayers, and M. Völkel, “Cool URIs for the Semantic Web,” W3C, Interest Group Note, Mar. 31, 2008. [Online]. Available: http://www.w3.org/TR/cooluris/ (cit. on p. 2).

[5] “223 Best Practices URI Construction,” W3C, Wiki, Mar. 14, 2012. [Online]. Available: http://www.w3.org/2011/gld/wiki/223_Best_Practices_URI_Construction (cit. on p. 2).

[6] P. Bryant, “REST-ful URI design,” 2PartsMagic Blog, May 30, 2012. [Online]. Available: http://blog.2partsmagic.com/restful-uri-design/ (cit. on p. 2).

[7] M. T. C. Benitez, “Best Practice for Web Data URI,” W3C, Editor’s Draft, Jun. 10, 2014. [Online]. Available: http://dragoman.org/duri/ed-1.html (cit. on p. 2).

[8] T. Kindberg and S. Hawke, “The ‘tag’ URI Scheme,” World Wide Web Consortium, Request for Comments 4151, Oct. 2005. [Online]. Available: http://tools.ietf.org/html/rfc4151 (cit. on p. 2).

[9] ——, (Jul. 9, 2008). Tag URI, [Online]. Available: http://www.taguri.org (cit. on p. 2).

[10] R. Moats, “URN Syntax,” IETF, Request for Comments 2141, May 1997. [Online]. Available: http://tools.ietf.org/html/rfc2141 (cit. on p. 2).

[11] L. L. Daigle, D.-W. van Gulik, R. Iannella, and P. Faltstrom, “URN Namespace Definition Mechanisms,” IETF, Request for Comments 2611, Jun. 1999. [Online]. Available: http://tools.ietf.org/html/rfc2611 (cit. on p. 2).

[12] P. J. Leach, M. Mealling, and R. Salz, “A Universally Unique IDentifier (UUID) URN Namespace,” IETF, Request for Comments 4122, Jul. 2005. [Online]. Available: http://tools.ietf.org/html/rfc4122 (cit. on p. 2).

[13] S. Harris, A. Seaborne, and E. Prud’hommeaux, “SPARQL 1.1 Query Language,” W3C, Recommendation, version 1.1, Mar. 21, 2013. [Online]. Available: http://www.w3.org/TR/sparql11-query/ (cit. on p. 3).

[14] E. Prud’hommeaux, G. Carothers, D. Beckett, and T. Berners-Lee, “RDF 1.1 Turtle, Terse RDF Triple Language,” W3C, Recommendation, version 1.1, Feb. 25, 2014. [Online]. Available: http://www.w3.org/TR/turtle/ (cit. on p. 3).

[15] D. Kolas, I. Emmons, and M. Dean, “Efficient Linked-List RDF Indexing in Parliament,” in *Proceedings of the Fifth International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2009)*, ser. Lecture Notes in Computer Science, vol. 5823, Washington, DC: Springer, Oct. 2009, pp. 17–32. [Online]. Available: http://ceur-ws.org/Vol-517/ (cit. on p. 3).

[16] D. Kolas, “Query Rewriting for Semantic Web Information Integration,” in *Proceedings of the Sixth International Workshop on Information Integration on the Web (IIWeb-07), at the Twenty-Second Conference on Artificial Intelligence (AAAI-07)*, Vancouver, Canada, Jul. 2007. [Online]. Available: http://www.aaai.org/Papers/Workshops/2007/WS-07-14/WS07-14-008.pdf (cit. on p. 3).

Comprehension of RDF Data Using Situation Theory and Concept Maps

Jakub J. Moskal
VIStology, Inc.
Framingham, MA 01701, USA
Email: jmoskal@vistology.com

Mieczyslaw M. Kokar
Northeastern University
Boston, MA 02115, USA
Email: m.kokar@neu.edu

Brian E. Ulicny
VIStology, Inc.
Framingham, MA 01701, USA
Email: bulicny@vistology.com

***Abstract*—The amount of RDF data available on the Web has been increasingly growing over the past few years. Developing and fine-tuning SPARQL queries in order to sift through the data may be a very challenging task for human operators who need to quickly make sense of large graphs. In addition, often multiple queries need to be issued in order to gather and understand the context (relevant facts) for the explanation of the query. Thus, the challenge is not only to answer the query, but also to provide context, so that the analyst can easily comprehend what the data is actually conveying.**

This paper describes results of an investigation of the possibility to apply key aspects of Situation Theory, and its ontological realization in the Situation Theory Ontology, to simplify and abstract large RDF data sets, given a focus query from the analyst. In this approach, the query results are presented as concept maps. The approach was successfully implemented as a prototype, although this paper does not include a description of the tool.

I. Introduction

Development of intelligence products in various domains, e.g., business or military, requires sifting through tremendously large amounts of data, most of which so far is in an unstructured (or semi-structured) form (text reports, web pages). This constitutes a very high challenge to the analyst who performs this kind of task. While the analyst has in mind an idea of the focus of the inquiry, the focus may exist only in the analyst's head and thus cannot be supported by a computer-based tool. One way for the analyst to tell the computer what is being looked for is to issue a search query, e.g., using keywords. However, the tools that support keyword-based text search will return documents (or pointers to) that contain the words; the analyst still needs to do the hard work of reviewing the plethora of documents returned. Another way is to first use a text processing tool that will analyze the documents, extract entities and relations identified in those documents and represent them in a structured language, e.g., Resource Description Language (RDF) [1], and then analyze the resulting formal representation using an appropriate query language. An example of the development in this domain is the idea of *Linked Data* [2], which has resulted, among others, in a quite large knowledge base called *DBpedia* [3].

In fact, DBpedia is just one of the numerous open datasets that have been published in RDF format. As the chart in Figure 1 shows, the number of such datasets has been rapidly growing in the recent years. Unfortunately, the RDF structured information is still very difficult to analyze. To illustrate the problem, consider an example of analyst query about a gang-related activity:

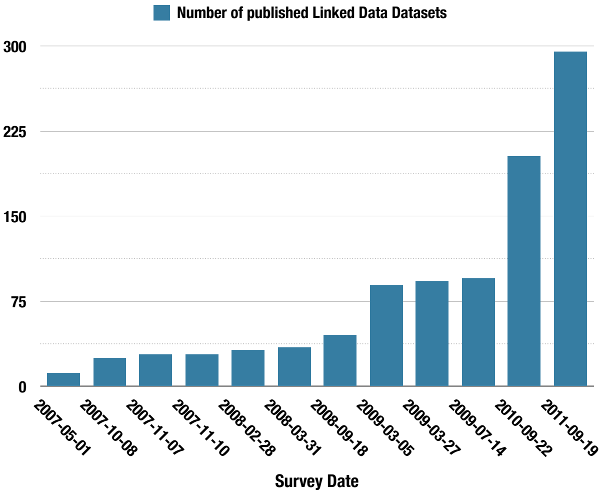


Fig. 1. Number of datasets that have been published in Linked Data format between 2007 and 2011 [4].

What were the circumstances of Richard H. Barter's death?

Such a query can be expressed in SPARQL query language [5] using the DESCRIBE query and the FILTER command that makes use of regex pattern matching to extract all the facts that are related to "Richard H. Barter". Even though DBpedia had only one resource ("Richard H. Barter") that is directly related to the query, the query returns more than 25 other resources that are one way or another related to this resource. DESCRIBE queries return RDF graphs and in order to analyze such an answer the analyst would have to go over all of the links and nodes and decide which of them are relevant.

Now the question is how to present the result of the query to the analyst? One of the formats for visual representation of complex information structures that has been proved quite successful in various uses, including knowledge structuring, learning and even knowledge creation, is the representation called *Concept Map* [6], [7], [8]. However, as discussed later in the paper, concept maps that are direct representations of

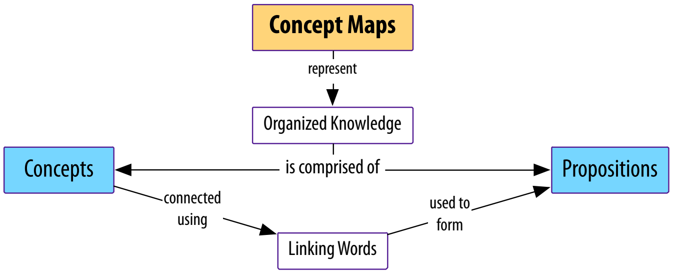


Fig. 2. Example of a concept map, representing the notion of concept maps itself [12].

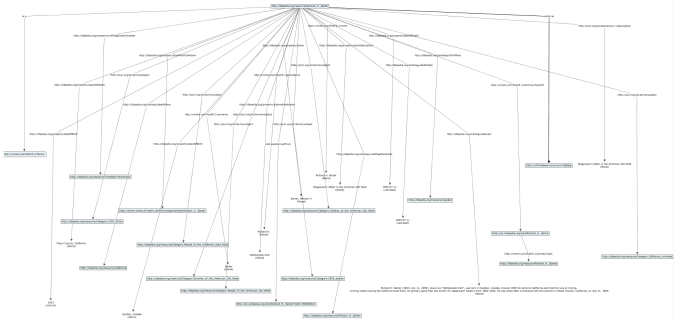


Fig. 3. Small RDF graph, returned by a SPARQL DESCRIBE query, visualized as a concept map using the IHMC CMap tool.

RDF graphs can also become quite complex and thus difficult to comprehend.

The problem addressed in this paper is the transformation of RDF to concept maps so that the resulting concept map is *relevant* to a specific analyst *query*, includes the appropriate *context*, and is presented in a more *abstract form* than the original RDF so that it is easy to comprehend. Our approach is to use key aspects of Situation Theory of Barwise and Perry [9], as extended and formalized by Devlin [10], map our problem to this theory and implement algorithms for constructing concept maps based on such a framework. In this work, we used the Situation Theory Ontology (STO) [11] that we developed earlier.

The rest of this paper is organized as follows. In Section II we briefly overview concept maps. In Section III we briefly discuss why Situation Theory is a good candidate for the solution. Then in Section IV we show how we can represent analyst queries in the STO ontology. This is followed by the discussion of domain inference in Section V and situation reasoning in Section VI. Section VII describes the derivation of (possibly) multiple contexts related to a query. Section VIII then discusses how the contexts are simplified in order to make the derived concept maps easier to comprehend. Finally, Section IX presents the conclusions of the paper and suggests some of the possible directions for future research.

II. Concept Maps

A *concept* is defined [8] as a perceived regularity or pattern designated by a label. *Propositions* are statements about some object or event in the universe, either naturally occurring or constructed. Propositions contain two or more concepts connected using linking words or phrases to form a meaningful statement. Sometimes these are called *semantic units*, or units of meaning.

Concept maps (c.f. Figure 2) include concepts (represented as boxes) and relationships between concepts (propositions) indicated by connecting lines linking pairs of concepts. Words in the boxes represent concept names, while words on/above the lines represent relationships between two concepts. Since concepts and properties are the building blocks of RDF, RDF graphs can be seen as concept maps. The CMap tools from IHMC can be used to provide graphical representations of RDF graphs as concept maps [8].

The concept map of Figure 2 shows five concepts and four propositions, where one of the concepts (Concept Maps) is a "meta-concept", since it represents the notion of a concept map itself. This map is only a fraction of a larger map, which shows the key features of concept maps [12]. Note that different look and feel styles can be applied to both concepts and linking words, e.g., different colors for different types of concepts.

Now returning to our example of the above described query, the CMap tool can load the answer to the SPARQL query and convert it to a concept map (c.f. Figure 3), however, the analysis of the map is still not that easy. One of the main reasons for this difficulty is the fact that concept maps generated in this way will contain too many concepts and relationships, many of them not relevant to the query. (Note: Clearly, Figure 3 is not readable. The sole purpose of this figure is to show the complexity of such concept maps.) One way to simplify the presentation would be to display just a small portion of the concepts and relationships. However, this operation needs to be performed very carefully so that important facts, without which the analyst would not be able to understand the answer to the query, are not omitted. Furthermore, the answer might include too detailed information, which clutters the global conceptual picture and defeats the purpose of the concept map. Hence, a fine balance between the simplicity and the amount of information must be kept in order to allow the analyst to quickly explore and understand the data.

III. Situation Theory

Situation Theory is "a set of mathematically-based tools to analyze, in particular, the way context facilitates and influences the rise and flow of information" [10]. Situation theory came about from the attempts to formalize *Situation Semantics* – reasoning about common sense and real world situations [9]. As postulated by Barwise and Perry, situations are first-class objects, i.e., they have their own existence, can stand in relation with other objects (including other situations) and can have their own attributes.

In situation theory, information about a situation is expressed in terms of *infons* written as:

$$\ll R, a_1, \ldots, a_n, 0/1 \gg$$

where R is an n-place relation and $a_1, \ldots, a_n$ are *objects* appropriate for R. Since situation theory is multi-sorted, the word "appropriate" means that the objects are of the types appropriate for a given relation. The last item in an infon is the *polarity* of the infon. Its value is either 1 (if the objects stand in the relation R) or 0 (if the objects don't stand in the relation R).

To capture the semantics of situations, situation theory provides a relation between situations and infons. This relationship is called the *supports* relationship which relates a situation with the infons that "are made factual" by the situation. Given an infon σ and situation s the *proposition* "s supports σ" is written as:

$$s \models \sigma.$$

The relation between a situation (in the world) and a representation of the situation (in a formal framework) is relative to a specific agent. It is the agent who establishes such a link. This link is defined by *connections* that link entities in the world to formal constructs of the situation-theoretic framework. These connections are not part of the formal theory. One refers to situations within a formal theory by using *abstract situations*, although the qualifier "abstract" is often dropped in most discussions of situation theory. An abstract situation is then a collection of infons supported by a specific situation.

In our approach we mapped key aspects of Situation Theory to Situation Theory Ontology (STO). The top-level classes of STO are shown in Figure 4. The details of this ontology were described elsewhere [13]. Here we just mention that the main idea behind this ontology is to capture the concept of "situation" (the Situation class serves this purpose). An individual s' of Situation that corresponds to a situation s in the real world, serves as the root to the description of the situation s. The abstract situation associated with s' is the *context*; it holds all the facts that are relevant to the situation, s. Other classes included in Figure 4 include Relation (to represent relations that individuals - instances of the class Individual - are involved in), Attribute (to represent attributes of both individuals and situations), Value and Dimensionality of the attributes, Rule (to represent rules for inferring higher-arity relations) and Polarity (to represent the values of Polarity; the only instances of this class are 1 and 0).

It is important to stress here that STO approximates Situation Theory by capturing the *supports* relation with a *entails* (or *derives*) relation, $\vdash$, between the collection of infons represening a situation and the infon representing a query [13]. Moreover, information is not represented in the form of infons. Instead, STO uses OWL and/or rules to represent knowledge about situations, i.e., abstract situations are captured by OWL sentences. However, as shown in Figure 4, STO includes the class ElementaryInfon. The sole role that ElementaryInfon plays in STO is to capture the focus of specific situations. I.e., queries (expressed in natural language) are formalized as instances of this class. ElementaryInfon resembles the structure of the infon in Situation Theory and thus has two types of properties: *relation* (to point to the relation, R that the infon represents), and *anchor* (not shown in Figure 4) to point to the arguments of R. Polarity in STO is represented explicitly, i.e., positive assertions correspond to polarity 1 and negative assertions correspond to polarity 0.

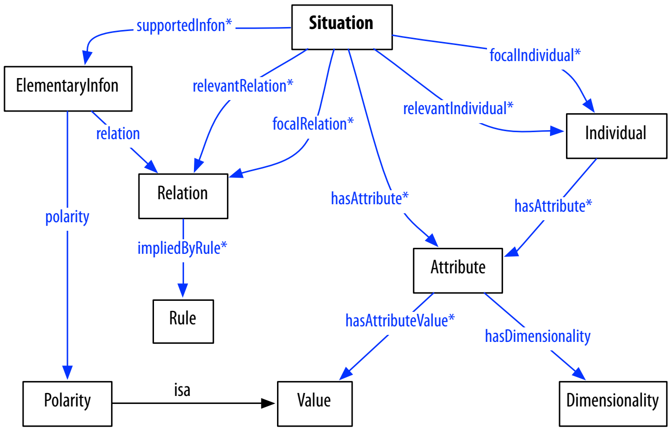


Fig. 4. Top-level classes in the STO ontology.

One of the possible alternatives to Situation Theory that we looked at was the FrameNet approach [14]. FrameNet is based on a theory of meaning called "frame semantics" derived from the work of Fillmore et al. (cf. [15]). The basic idea is that the meanings of most words can best be understood on the basis of a semantic frame: a description of a type of event, relation, or entity and the participants in it.

While this idea seems to be close to that of the Situation Theory (ST) semantics, the latter has a number of advantages that make ST a better match for this particular problem. (1) ST grounds meaning in the world rather than in the language. This allows for the development of situation types that have meaning in the physical reality (e.g., battlefield, ships, missiles and so on), and not just in the syntax of the human language. (2) Unlike in more pure logic-based semantics, meaning in ST is provided by partial views of the world, not all possible worlds. This gives an advantage of being able to specify views of the world (situations) that are globally inconsistent, but locally consistent. This will allow analysts to specify situation types that, when taken together, are inconsistent. This capability allows the deference of the resolution of inconsistencies to the interactions with the world, rather than trying to develop a consistent set of types (an impossible state to achieve) before anything is utilized. (3) Situations in ST are first-class objects, i.e., they not only stand in relations with other situations, but can have their own attributes and properties. (4) In ST the meaning of a declarative sentence is a relation between utterances and described situations, which is exactly what is needed for a solution to our problem — developing concept maps that support the understanding of answers to specific analyst's questions (queries).

IV. Representing Queries

In our approach, the essence of the textual version of analyst queries needs to be extracted and mapped to to the ontology. Since situations are explicitly represented in STO, the mapping of the queries to STO has to be consistent with the intent of this ontology. In particular, since the intent is to connect a query with a context (which in STO is captured by a situation), as well as to ensure that the relevant facts are included in the context, queries were mapped to the class of ElementaryInfon and to a specific situation type.[1] For instance, the answer to the query whose textual representation is

“Did an insurgent visit a weapons cache”?”

can be captured by InsurgentWeaponsCacheSituation (a subclass of the Situation class), defined in OWL as follows:

InsurgentWeaponsCacheSituation ≡ Situation **and** (supportedInfon **some** (ElementaryInfon **and** (anchor1 **some** Insurgent) **and** (anchor2 **some** WeaponsCache) **and** (relation **value** visit)))

Answering such a query would involve inferring whether the current knowledge base supports the conclusion that there is a situation individual that is a member of the class InsurgentWeaponsCacheSituation. Note that the above definition assumes that the domain-specific ontology used in this query extends STO with some classes (e.g., Insurgent, WeaponsCache) and relations (e.g., visit).

Unfortunately, OWL is not sufficient enough to express some types of queries. For instance, the following query cannot be expressed in OWL alone:

“Which insurgents spied on a relative?”

The reason for this is that one needs to refer to variables, which are not supported by OWL. In particular, the intent of this query is to identify only those insurgents who spied on their own relatives, not just any insurgents who spied someone’s relatives. In such cases one needs to use rules. For instance, using the STO, the query above could be expressed as the following rule:

Situation(s) ∧ ElementaryInfon(i) ∧ Object(a1) ∧ Object(a2) ∧ Relation(spiedOn) ∧ supportedInfon(s,i) ∧ anchor1(i, a1) ∧ anchor2(i, a2) ∧ relation(i, spiedOn) ∧ Insurgent(a1) ∧ Person(a2) ∧ relative(anchor1, anchor2) → RelativeSpySituation(s)

Such rules can be captured in SPARQL 1.1 (using INSERT to assert new facts) or in an inference engine-specific language like BaseVISor’s RDF-based BVR [16]. For the ease of use, since it was already the language in which some of the domain axioms were expressed (discussed below), BVR was chosen as the query language. In BVR, rules are defined within a rule base with each rule consisting of a *body* element and a *head* element (which can occur in either order). The name attribute can be used to assign a name to a rule base or rule. The heads and the bodies use the *triple syntax*, i.e., each rule consists of clauses, each being a triple (predicate, subject, object). The syntax of BVR is conceptually compatible with RDF. This kind of rules are easy to write and interpret; the only problem is that it is verbose. For this reason, BVR offers an abbreviated syntax [16].

The activities involved in the answering of analyst queries and creating concept maps that constitute the answers, is shown in Figure 5. The following sections describe each of these activities in more detail.

V. Domain Inference

The first step in the processing of an analyst query is to run the inference on the supplied RDF data and infer implicit facts about the domain (Step 1 in Figure 5). Since RDF does not provide strong axioms for inference, the RDF data can be augmented with additional axioms expressed in OWL and rules. OWL was the preferred choice, but if for some axioms it was not expressive enough, axioms were added in the form of BVR rules.

For instance, for the SynCOIN dataset [17] used in our experiments, examples of domain-specific axioms are definitions of object properties *associate* and *madeTransactionsWith*, both of which were defined as sub-properties of the transitive and symmetric *isConnectedTo* property (left side of Figure 6). An example of the use of these axioms is shown on the right side of the figure. Assuming that only John has been to a weapons cache, and that Mary is the only known insurgent, if the analyst issues a query *“Which known insurgents are connected to people who have been to a weapons cache?”*, the system should produce a map that includes Mary and John. In addition, the map should also include Bob and the relationships between all individuals, in order to fully represent the context. Without Bob in the result, it is not obvious how Mary and John are actually connected.

While the process of adding domain-specific axioms needs to be done manually, it is part of the knowledge engineering task, which is expected to be performed for each domain of application. Obviously, automatic ways of generating such axioms are desirable, but this was not part of this investigation. In our case, we arbitrarily decided which axioms to include.

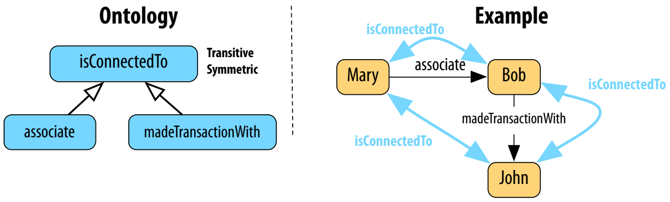


Fig. 6. Property taxonomy and example of its use. The blue lines represent implicit, inferred facts.

VI. Situation Reasoning

Once the domain inference is complete and all implicit domain facts are asserted in the knowledge base, individuals of a situation type that corresponds to the query, as well as relations among them, can be found (Step 2 in Figure 5).

[1] In OWL a query about some individuals can be viewed as a class, i.e., a collection of those individuals that satisfy the definition of the class.

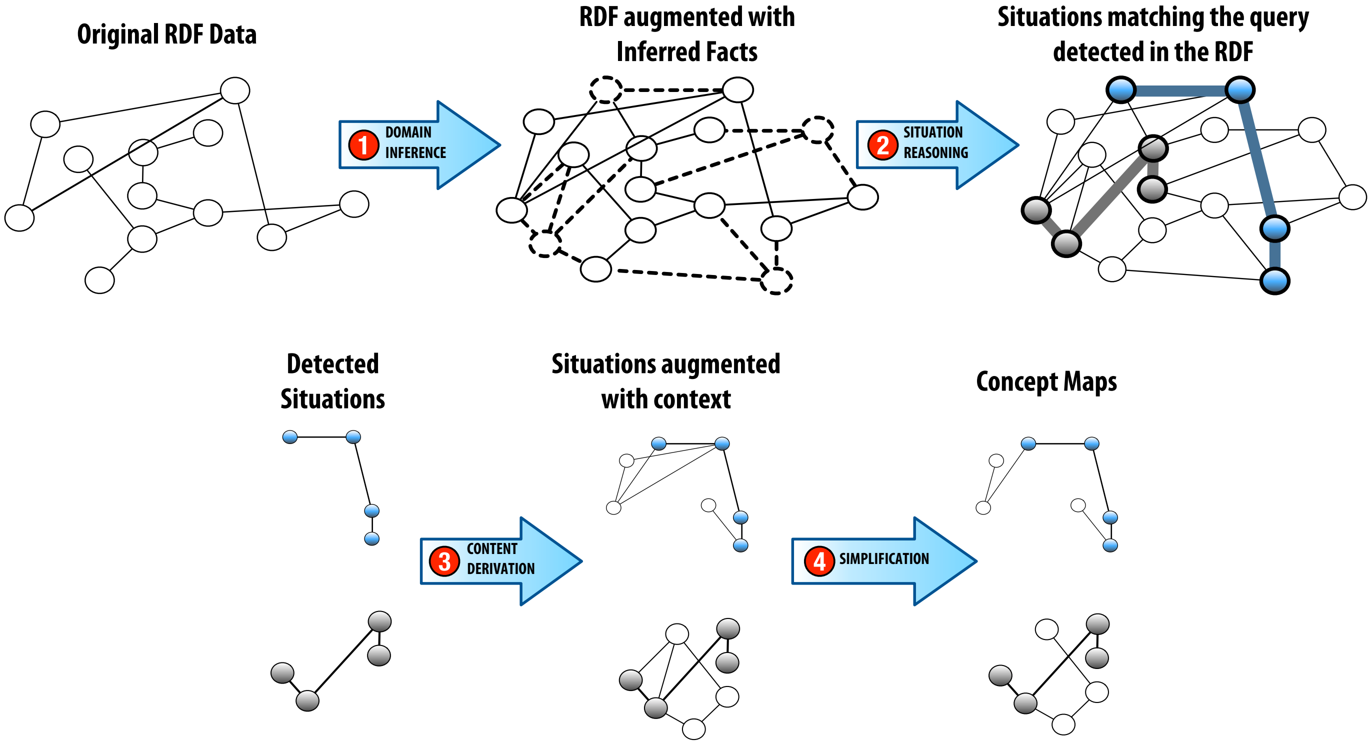

Original RDF Data
DOMAIN INFERENCE
RDF augmented with Inferred Facts
SITUATION REASONING
Situations matching the query detected in the RDF
Detected Situations
CONTENT DERIVATION
Situations augmented with context
SIMPLIFICATION
Concept Maps

Fig. 5. The process of transforming RDF data into comprehensible concept maps described in this paper. The situations detected in the RDF graph correspond to answers to an analyst query, which gives a focus for the produced concept maps.

To begin with, situation type definitions need to be analyzed — both those that are defined in pure OWL and those that are defined in rules (see Section IV for details). The main focus here is to extract the relations used in the definition of the situation types. For instance, for the InsurgentWeaponsCacheSituation type, the system should extract the *visit* relation. Similarly, for RelativeSpySituation, it would extract the *spiedOn* relation. Getting this information from OWL definitions is trivial, since we know the structure of the definition of situation types, which use the notion of ElementaryInfon, which in turn explicitly uses the object property *relation*. It gets more complicated with the situation types defined in rules. In our experiments, the BVR rule files were processed with regular expressions in order to find the relations. In the future, the rules themselves could be formally represented in OWL and the solution could avoid the use of regular expressions. Once the relations from situation type definitions are known, the process of asserting situation individuals is as follows:

- For each relation rel that is part of a situation type:
  - For each pair of individuals a_1 and a_2 that are associated with each other by the property rel:
    1) Assert that there is an individual s of RDF type sto:Situation
    2) Assert that there is an individual i of RDF type sto:ElementaryInfon, supported by situation s
    3) Assert the following facts: (i $anchor1$ a_1), (i $anchor2$ a_2) and (i $relation$ rel)

Now the reasoner can infer the situation types of the situation individuals.

VII. Derivation of Contexts

At this point, the answers consist of the anchors and the relations used in the situation definitions. For instance, for the weapons cache query and the axioms shown in Figure 6, given that John has been to a weapons cache, the system would return a basic concept map including Mary, John and isConnectedTo, but not include Bob and his relationships with them, which would explain why Mary is actually connected to John. Hence, the next processing step is to derive the context for the answer, i.e., find all individuals and relations that are relevant to the situation that represents the answer to the query. This corresponds to step 3 in Figure 5. Recall that “context” means an abstract situation, as described earlier in the paper. The main idea is that context is the description of a situation, including all the relevant individuals and the relevant relations among the individuals. All of this (the context) is captured by the *relevant facts*, i.e., facts that assert which individuals and relations are relevant and what are the relations among the relevant individuals.

For deriving context, we implemented a set of domain-independent rules, that backtrack some of the OWL inference rules. For instance, if a relation that is relevant to the query is defined as a property chain, the individuals and relations that form the chain are inferred to be relevant as well. Similarly, if a relevant relation is defined as a super-property of another property that holds between two relevant individuals, it is also inferred to be relevant. At the time of writing, the set of the context derivation rules is not complete, i.e., not every OWL inference rule that produces new facts has a corresponding relevance derivation rule. Also, some rules might produce facts that are not necessary to explain a situation to the analyst, thus producing some "noise". Such issues are on our agenda for future work.

As an example, the following describes one of the relevance derivation rules related to the transitive properties in OWL[2]:

```
• For a situation $s$, and a query $q$, if $s$ satisfies the query:
    – For every fact $(i_1\ rel\ i_2)$ relevant to $s$ and an individual $i_3$, if $rel$ is a transitive property and if $(i_1\ rel\ i_3)$ and $(i_3\ rel\ i_2)$ are facts asserted in the knowledge base:
        1) Add $(i_1\ rel\ i_3)$ and $(i_3\ rel\ i_2)$ as facts relevant to $s$.
```

Figure 7 shows how derivation rules can be applied in the weapons cache example, given the axioms in Figure 6. First, based on the above rule applied to *isConnectedTo*, the inference engine would infer that the individual Bob is also relevant and should be part of the context (Figure 7b). Moreover, using a different derivation rule, the reasoner would infer that *associate* and *madeTransactionWith* are also relevant, because they are sub-properties of a relevant property and hold between relevant individuals (Figure 7c).

Note that not only individuals and properties are asserted as relevant to a situation, but entire facts (triples) are also asserted as such. It is not sufficient to just list the individuals and properties without showing the associations between them. In our experiments, we used the notion of OWL annotation properties in order to annotate facts as relevant to specific situation individuals. Since OWL does not support reasoning over annotation properties, the only way to implement such reasoning is to use rules. As we mentioned earlier in the paper, our preference was to use OWL reasoning first and add rules only out of necessity.

VIII. Simplification of Concept Maps

One can easily see that as a result of context derivation reasoning, the number of relevant facts for each situation might grow fast and if converted into a concept map, it could look quite convoluted (compare Figure 7a with Figure 7c). More importantly, it would most likely include redundant facts. For instance, Figure 7c shows that Mary and Bob are associated using two properties *isConnectedTo* and *associate*, although the former is just a generalization of the latter.

In order to make such resulting concept map less cluttered, and thus easier to comprehend, we need to remove facts that are relevant to a situation, but that are not necessary to comprehend the graph. We call this step *context simplification* and it corresponds to step 4 in Figure 5.

Similarly to the previous steps, for this purpose we developed a number of domain-independent rules that remove redundant facts. As an example, the following algorithm describes the rule that removes from a situation's context those properties whose sub-properties, holding between the same individuals, are relevant, yet not necessary:

```
• For a situation $s$, and a query $q$, if $s$ satisfies the query:
    – For every relation $r_1$ and $r_2$ both relevant to $s$, if $r_1$ is a sub-property of $r_2$:
        ∗ For every two facts $(i_1\ r_1\ i_2)$ and $(i_1\ r_2\ i_2)$ that are both relevant to $s$:
            1) Remove $(i_1\ r_2\ i_2)$ from the context of $s$.
```

Back to the weapons cache example, based on the above rule applied to the graph in Figure 7, the system would remove the two *isConnected* links between Bob and the other two people, since they both provide redundant information. The *associate* and *madeTransactionWith* properties are more specific and clearly explain the context for the original query.

The resulting concept map could use different graphical styles when rendering concepts and links, in order to distinguish the query answer itself from its context. This approach gives the analyst a quick focus on the most important concepts in the graph, but also provides the context without cluttering the answer.

IX. Conclusion

The main objective of the research described in this paper was to investigate the possibility of using the ideas from Situation Theory (Barwise, Perry and Devlin), and its ontological realization in the Situation Theory Ontology, to the task of simplifying and abstracting concept maps, provided as RDF graphs, so that they are easier to comprehend by an analyst while still preserving the semantics of the original representation. This paper covers only some of the aspects of this investigation. In particular, it shows (by example) how an analyst's query can be mapped to an ontological representation, what it takes to derive facts that are relevant to the query, and how to represent such facts in graphical form (both with and without auxiliary facts that provide an explanation to the analyst of how they were derived). This investigation ended with a prototype tool (not included in this paper) for generating, displaying and manipulating concept maps in order to improve their comprehensibility. The next logical task for this research is to evaluate the tool on a representative number of queries and datasets and assess the approach with respect to its completeness and the strength of the rules used for the simplification of the query results. In particular, such an evaluation would require human-in-the-loop, i.e., the involvement of the analysts performing analyses of situational awareness in their domains.

Acknowledgment

This work was performed under Office of Naval Research contract N00014-14-P-1081 "Concept Maps from RDF (Resource Description Framework)". Any opinions, findings and

[2] Note that it is not important whether the facts on which the rule operates were derived or asserted by the user.

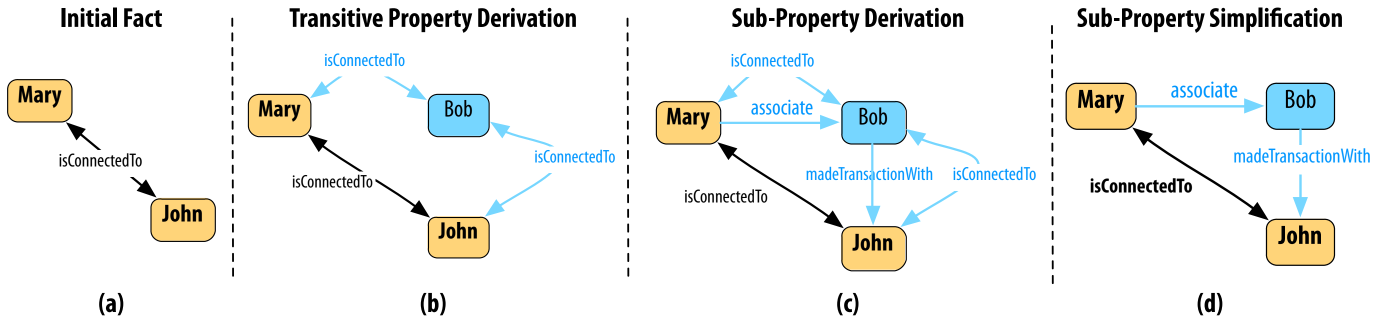


Fig. 7. Example of the context derivation and simplification of a query answer rendered as a concept map.

conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Office of Naval Research. The authors would also like to thank the anonymous reviewers who provided many constructive suggestions for improving the presentation and for future research directions.

References

[1] W3C, “RDF semantics. W3C recommendation 10 february, 2004,” Feburary 2004. [Online]. Available: http://www.w3.org/TR/2004/REC-rdf-mt-20040210/

[2] T. Heath and C. Bizer, *Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web: Theory and Technology*. Morgan & Claypool Publishers, 2011.

[3] DBPedia-Community, “DBpedia,” September 2014. [Online]. Available: http://dbpedia.org/About/

[4] R. Cyganiak and A. Jentzsch, “The linked open data cloud diagram,” http://lod-cloud.net/, September 19 2011.

[5] J. Pérez, M. Arenas, and C. Gutierrez, “Semantics and complexity of sparql,” in *The Semantic Web-ISWC 2006*. Springer, 2006, pp. 30–43.

[6] J. D. Novak, “Concept maps and vee diagrams: Two metacognitive tools for science and mathematics education,” *Instructional Science*, vol. 19, pp. 29–52, 1990.

[7] E. Plotnik, “Concept Mapping: A Graphical System for Understanding the Relationship between Concepts,” *ERIC Clearinghouse on Information and Technology Syracuse NY*, vol. ED407938, 1997.

[8] J. D. Novak and A. Cañas, “The Theory Underlying Concept Maps and How to Construct and Use Them,” cmap.ihmc.us/publications/researchpapers/theorycmaps/.

[9] J. Barwise and J. Perry, *Situations and Attitudes*. Cambridge, MA: MIT Press, 1983.

[10] K. Devlin, “Situation theory and situation semantics,” in *Handbook of the History of Logic*, D. M. Gabbay and J. Woods, Eds. Elsevier, 2006.

[11] “Situation Theory Ontology.” [Online]. Available: http://vistology.com/onts/2008/STO/STO.owl

[12] J. D. Novak and A. J. Cañas, “The theory underlying concept maps and how to construct and use them,” Institute for Human and Machine Cognition, Tech. Rep. Technical Report IHMC CmapTools 2006-01 Rev 2008-01, 2008. [Online]. Available: http://cmap.ihmc.us/publications/researchpapers/theorycmaps/theoryunderlyingconceptmaps.htm

[13] M. M. Kokar, C. J. Matheus, and K. Baclawski, “Ontology-based situation awareness,” *Information fusion*, vol. 10, no. 1, pp. 83–98, 2009.

[14] C. F. Baker, C. J. Fillmore, and J. B. Lowe, “The Berkeley FrameNet project,” in *Proceedings of COLING/ACL*, 1998.

[15] C. J. Fillmore, “Frame semantics,” in *Linguistics in the Morning Calm*. Seoul, Korea: Hanshin Publishing Co., 1982, pp. 111–137.

[16] C. J. Matheus, K. Baclawski, and M. M. Kokar, “BaseVISor: A triples-based inference engine outfitted to process RuleML and R-Entailment rules,” in *Rules and Rule Markup Languages for the Semantic Web, Second International Conference on*, 2006, pp. 67–74.

[17] J. L. Graham, D. L. Hall, and J. Rimland, “A coin-inspired synthetic dataset for qualitative evaluation of hard and soft fusion systems,” in *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*. IEEE, 2011, pp. 1–8.

Toward the Discovery and Extraction of Money Laundering Evidence from Arbitrary Data Formats using Combinatory Reductions

Alonza Mumford, Duminda Wijesekera
George Mason University
amumford@gmu.edu, dwijesek@gmu.edu

***Abstract*—The evidence of money laundering schemes exist undetected in the electronic files of banks and insurance firms scattered around the world. Intelligence and law enforcement analysts, impelled by the duty to discover connections to drug cartels and other participants in these criminal activities, require the information to be searchable and extractable from all types of data formats. In this overview paper, we articulate an approach — a capability that uses a *data description language* called Data Format Description Language (DFDL) extended with higher-order functions as a host language to XML Linking (XLink) and XML Pointer (XPointer) languages in order to link, discover and extract *financial data fragments* from raw-data stores not co-located with each other —see figure 1. The strength of the approach is grounded in the specification of a declarative compiler for our concrete language using a higher-order rewriting system with binders called Combinatory Reduction Systems Extended (CRSX). By leveraging CRSX, we anticipate formal operational semantics of our language and significant optimization of the compiler.**

***Index Terms*—Semantic Web, Data models, Functional programming, Data processing, Formal languages, Law enforcement**

I. Introduction

The approach leverages emerging developments in *data description languages* such as Data Format Description Language (DFDL) [1] for providing efficient representations of dense binary and textual data formats through vendor-neutral mechanisms. A DFDL schema allows raw data to be read from its native data format as an instance of a DFDL data model, and equivalently, composed to raw data from an instance of a DFDL data model. Within the context of this application, a DFDL schema represents a data repository containing any data format because the schema can refer to the local storage of the data it describes and provide instructions as to how that data may be read or written in its native form (e.g., bits, formats). Further, the concept outlines the addition of new abstractions to DFDL for defining the relationship and linkage between data fragments corresponding to different data files as well as for functions for extracting data fragments. The strength of this approach is grounded in the specification of a declarative parser-generator for these DFDL extensions using CRSX, which implements Klop's Combinatory Reduction Systems (CRS) with extensions to support the writing of compilers [2].

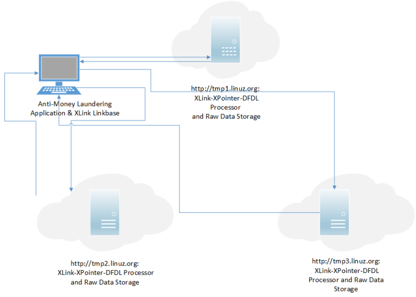


Fig. 1. An illustration of an anti-money laundering application that connects to multiple data storage sites. In this case, the native data format at each site differs, and a *data description language* extended with higher-order functions and linking/pointing abstractions are used to extract *data fragments* based on their ontological meaning.

II. Related Work

This work presents a multifaceted challenge that primarily breaks out into two areas. The first challenge is to provide a mechanism that can be used to describe and access any number of data formats. A class of data parsing languages commonly referred to as *data description languages* to include PADS[3] and DATASCRIPT have demonstrated this potential. This capability is not the same as offered by prescriptive data format languages such as JSON or even JSON-LD [4], which require compliance to a pre-specified structure and physical format. Descriptive languages have the advantage of being able to describe a data model's logical representation, which defines the semantics of the data, as well as its physical representation, which defines the methods by which its stored, without having to alter the target data from its initial format.

The second challenge is to combine the former capability with a lightweight-mechanism that supports metadata-based discovery and extraction of arbitrary data fragments from raw data stores without the system development and maintenance costs associated with major data conversion, and database storage and indexing. While popular data storage and extraction schemes such as Apache's Hadoop/MapReduce [5] and Accumulo/Big Table [6] provide a rich software-framework, they typically require data conversion for querying data fragments.

Unlike its *data description language* cohorts, DFDL extends a subset of XML Schema Description Language (XSDL), and augments the inherit logical model of the schema with DFDL annotations that are used to describe the physical representation of the data. In the same manner, our approach further extends DFDL's logical model with annotations for semantic-based traversal between local and remote resources that can be used to facilitate distributed discovery, parsing and extraction of raw data fragments. In addition to being interpretable by external XLink-XPointer processors, these semantic annotations also serve as instructions to the DFDL compiler for parser generation.

III. Methodology

The high-level methodology for this research proposal has been decomposed into four components. First, a plausible money laundering scheme is provided, and some inferences regarding how that scenario may appear in financial information is conceived. Second, the data records that are the focus of the money laundering investigation are examined for their format and subsequently, the data is modeled to understand its logical and physical specification. Third, the information within the data is conceptualized based on the "money scheme" and modeled for its semantic relationships using linking and pointing abstractions. Fourth, parts of the specification for the parser-generator are provided. Further details are provided in each respective section of the paper.

IV. Analysis of a Money Laundering Scheme

Consider an example of a money laundering scheme that could be used by any analyst to drive the discovery and extraction of data with evidentiary value from electronic insurance records. In the example scheme, the early cancellation of insurance policies with *return of premiums* has been used to launder money. Based on an assessment from the law enforcement community, this kind of insurance scheme has occurred where there have been [7]:

- A number of policies entered into by the same insurer (i.e. a person or company that underwrites an insurance risk) for small amounts and then canceled at the same time
- Return premium being credited to an account different from the original account, and
- Requests for return premiums in currencies different to the original premium.

V. Data Modeling

A. Inspection of the Insurance Account Records

The information referenced in the *return premium* scheme is made available in three electronic insurance records: the *policyAccountRecord*, *cancelRequestRecord*, and *creditRequestRecord*. At this step, the analyst's objective is to inspect the data such that he may describe a data model for each format type, which consist of a logical structure and physical representation.

In figure 2(a.1), the logical structure of the *policy AccountRecord* is a sequence complexType named by the identifier *PLCYACCT*. The *policyAccountRecord* type can be viewed as a data structure, where its value contains other values and its definition contains a datatype and identifier for each field. For example, *INSUR* is the identifier for a simpleType field named *Insurer*. The physical representation of the *policyAccountRecord* type is also describable. Delimiters, which are a sequence of characters, are used to specify the boundary between separate, independent areas in the text representation. For example, "/" is an infix separator between an identifier and value such as *PAYCUR* and *Peso (ARG)*, and white-space is a initiator and "//" is a terminator for each field. Also, the character-encoding scheme for the text representation is identified as ASCII. In figure 2(b.1), the *creditRequestRecord* type is given using some peculiar characters for separating the fields in the record. The *cancelRequestRecord* type uses the standardized JSON format (not shown). At this stage, the analyst discovers that each of the record types do not share a common format type such as XML or JSON.

B. DFDL-based Data Modeling for Parsing

At this step, the analyst models the logical model in the sequential order of the data file using the "logical datatypes & constraints" such as those in listed in figure 3. Then, the analyst maps the physical representation of each data file to its logical model, using the "physical representation properties" like those also shown in figure 3.

At compilation, a *DFDL schema* model generates a "program," which is essentially a parser and unparser. Upon parsing, if the input *policyAccountRecord* data file (ref: 2(a.1)), for example, satisfies all the constraints specified by the *policyAccountRecord.dfdl* schema file (ref: figure 4), it is considered to be valid according to the schema. More importantly, the DFDL parser generates a logical representation of that input data file, shown in figure 2(a.2), called the DFDL information set (infoset) or data model. In return, the DFDL information set can be used to unparse or generate a data file.

VI. Ontological Modeling

A. Inspection of the Application Domain Context

At this point, the analyst will apply his analytical reasoning to define the concepts that are relevant to the money laundering domain. In *ontological engineering*, a *concept* definition conveys the name of an evidentiary fact and its value data type. In figure 5, the generic kinds of conceptual abstractions are given along with corresponding examples of how those abstractions are applied within the example domain. In figure 6, a conceptual ontology of the "anti-money laundering" domain is given to show the kinds of classes and properties used in the domain. Generally, classes are identified by nodes and properties are identified by directed paths or *arcs*. This figure illustrates, for example, how these conceptual labels such as *creditRequestRecord* are structured taxonomically by composition (i.e., *hasPart*) and equivalence (i.e., *sameAs*) relations.

DFDL Logical Datatypes & Constraints (a)	**Context Free Grammar (CFG) example**	**Higher-Order Abstract Syntax (HOAS) example**
Structures (xs:complexType)	2 <dcl_xs> ::= "< " XS_COMPONENT <stmt>* "</" XS_COMPONENT ">" 11 XS_COMPONENT ::= "xs:complexType"	XsComponent[ComplexType]
Atomic data values (xs:simpleType)	11 XS_COMPONENT ::= "xs:simpleType"	XsComponent[SimpleType]
Ordering (xs:sequence or xs:choice)	11 XS_COMPONENT ::= "xs:sequence" | "xs:choice"	XsComponent[Sequence], XsComponent[Choice]
Occurences (xs:minOccurs or xs:maxOccurs)	5 <stmt> ::= XS_ATTRIBUTE "=" <xs_attribute_value> 16 XS_ATTRIBUTE ::= "xs:minOccurs" | "xs:maxOccurs" 8 <xs_attribute_value> ::= <xs_number>	ComponentAttribute[MinOccurs], ComponentAttribute[MaxOccurs]
DFDL Physical Representation Properties (b)		
Physical types (dfdl:representation)	5 <stmt> ::= DFDL_ATTRIBUTE "=" <dfdl_attribute_value> 18 DFDL_ATTRIBUTE ::= "representation" 7 <dfdl_attribute_value> ::= <dfdl_enum_value>	FormatProperty[Represetation]
Delimiters (dfdl:initiator, dfdl:separator, dfdl:terminator)	18 DFDL_ATTRIBUTE ::= "initiator" 7 <dfdl_attribute_value> ::= <dfdl_string_value> | <reg_exp_value>	FormatProperty[Initator], FormatProperty[Separator], FormatProperty[Terminator
Extraction of elements (dfdl:lengthKind)	18 DFDL_ATTRIBUTE ::= "lengthKind" 7 <dfdl_attribute_value> ::= <dfdl_enum_value>	FormatProperty[LengthKind]
Points of uncertainty (dfdl:discriminator)	3 <dcl_dfdl> ::= "<" DFDL_ADMIN <stmt>* ">" <dcl_dfdl> "</" DFDL_ADMIN ">" 15 DFDL_ADMIN ::= "dfdl:discriminator"	DfdlValidation[Discriminator]
Detecting occurrences (dfdl:occursCount)	18 DFDL_ATTRIBUTE ::= "occursCount" 7 <dfdl_attribute_value> ::= <non_neg_int_value> | <dfdl_exp_value>	FormatProperty[OccursCount]
XLink Properties (c)		
XLink type and label attributes (xlink:type or xlink:label)	5 <stmt> ::= XLK_ATTRIBUTE "=" <xlk_attribute value> 23 XLK_ATTRIBUTE ::= "xlink:type" | "xlink:label" 89 <xlk_attribute_value> ::= <xlktype_enum>	XlinkAttribute[XlkType], XlinkAttribute[XlkLabel]
DFDL Higher-Order Functions (HOF) (d)		
DFDL hof (dfdl_ext:filter or dfdl_ext:contains)	19 DFDL_HOF ::= "dfld_ext:filter" | "dfdl_ext:contains"	DfdlHof[Filter], DfdlHof[Contains]

Fig. 3. A specification of a DFDL compiler using CRSX performs *stepwise* transformations from the DFDL *concrete syntax* (shown in left column)) to an equivalent *higher-order abstract syntax* (HOAS) intermediate language (in right column). This transformation to the target language matches a context-free grammar (CFG) syntactic rule (in center column) to each unit of DFDL syntax and uses CRS-based *rewrite rules* to address semantic and optimization concerns.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:dfdl="http://www.ogf.org/dfdl/dfld-1.0/"
     xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:dfdl_ext="http://linuz.org/dfdl_ext" xlink:type="
     extended">
    ...
    <xs:element name="policyAccountRecord" minOccurs="0" maxOccurs="unbounded" dfdl:lengthKind="implicit
         " xlink:type="resource">
       <xs:complexType>
             <xs:sequence dfdl:sequenceKind="ordered">
                <annotation>
                   <xs:appinfo source="http://www.ogf.org/dfdl/v1.0">
                      <dfdl:element representation="text" encoding="ascii" lengthKind="delimited"
                           sequenceKind="ordered" initiator="//" separator="/" separatorPosition="infix"
                           separatorPolicy="required"/>
                      <dfdl_ext:filter>
                         <dfdl_ext:param def=[function_definition]/>
                      </dfdl_ext:filter>
                   </xs:appinfo>
                </annotation>
                <xs:element name="policyAccountIdentifier" type="xs:string" dfdl:lengthKind="explicit"
                     dfdl:length="20" xlink:type="resource" xlink:href="http://linuz1/policyAccountSchema
                     .dfdl#xpointer(///policyAccountIdentifier[@xs:string=value])"/>
                <xs:element name="policyStartDate" ... xlink:label="policyStartDate" xlink:type="resource
                     " xlink:href="http://linuz1/policyAccountSchema.dfdl#xpointer(///policyStartDate[
                     @xs:date=value])"/>
                <xs:element name="policyHolder" ... xlink:label="policyHolder" xlink:type="resource"
                     xlink:href="http://linuz1/policyAccountSchema.dfdl#xpointer(///policyHolder[
                     @xs:string=value])"/>
                <xs:element name="policyInsurer" ... xlink:label="policyInsurer" xlink:type="resource"
                     xlink:href="http://linux1/policyAccountSchema.dfdl#xpointer(///policyInsurer[
                     @xs:string=value])"/>
                <xs:element name="payerName" ... xlink:label="payerName" xlink:type="resource" xlink:href
                     ="http://linuz1/policyAccountSchema.dfdl#xpointer(///payerName[@xs:string=value])"/>
                <xs:element name="payerCurrency" ... xlink:label="payerCurrency" xlink:type="resource"
                     xlink:href="http://linuz1/policyAccountSchema.dfdl#xpointer(///payerCurrency[
                     @xs:string=value])"/>
                <xs:element name="premiumAmount" ... xlink:label="premiumAmount" xlink:type="resource"
                     xlink:href="http://linuz1/policyAccountSchema.dfdl#xpointer(///premiumAmount[
                     @xs:string=value])"/>
             </xs:sequence>
          </xs:complexType>
      </xs:element>
       ...
</xs:schema>
```

Fig. 4. The *Policy Account Record* DFDL schema illustrates attributes and elements belonging to the XLink (e.g., *xlink:type*) and extended DFDL (e.g., *dfdl_ext:filter*) namespaces. Note that the DFDL *name* attribute (e.g., *name="premiumAmount"*) is a named reference type to the *data model* context, while the *xlink:label* attribute is a named reference type to the *ontological model* context as defined through XLink and anti-money laundering application domain. Therefore, if the application domain shifts to a new domain of inquiry, then a new schema with the same data model but different conceptual labels can be devised.

B. XLink-XPointer-based Concept Modeling for Data Linking, Addressing & Extraction

XLinks can be embedded within a XML document that contains links between other XML or non-XML documents. Since any DFDL schema is also an XML document, XLinks can be placed within DFDL schemas. In figures 4 and 7, both *policyAccountRecord.dfdl* and *moneyLaunderLinkbase.xml* specify an *extended link*, which defines a collection of resources and a collection of arcs between resources. Not only does each resource point to a *financial data fragment*, it also represents a *concept* within the *anti-money laundering ontology* shown in figure 6. In figure 4, *resource* attributes mark local resources. In this case, a local resource is equivalent to a locally-stored data fragment that can be read from or written to by an associated DFDL schema. Each remote DFDL schema contains multiple XLink annotations in order to identify the respective data fragment's location.

In order to create a connection between two resources and define the *meaning* of the relationship between them, an *anti-money laundering* linkbase is devised —see figure 7. A linkbase [8] provides the *location* and *label* information for each *financial data* resource. A linkbase describes links between resources by providing an *arc* defintion or traveral instruction. On line 12, for example, the traversal from the source resource, *linkbase*, to the destination resource, *http://tmp1.linuz.org/policyAccountRecord.dfdl* is defined. The XLink syntax grants for a number of attributes in the XLink namespace, as shown in figure 8.

For data fragment addressing and selecting, XPointer expressions are applied in the xlink:href attributes of various

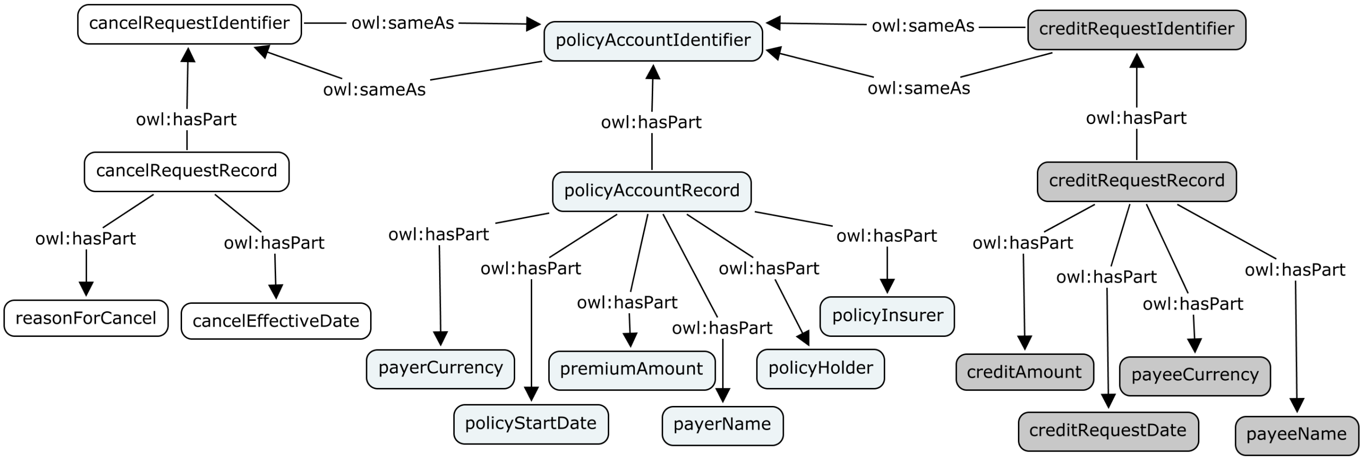


Fig. 6. The semantic relationships between the various concepts (data fragments) involved in a money laundering scheme may be illustrated in our ontological model of the domain.

elements in order to point to the data fields of the three data record types. In figure 4 on line 6, for example, the link address the *reasonForCancel* element with *xs:string* value is provided. In the example, XPointer depends on XPath expressions to point to resources. This *data linking* architecture is defined by a linkbase as well as corresponding XLink/XPointer-extended DFDL schemas for each record type and remote data stores.

VII. Extensions for Higher-Order Functions

This section focus on the utility of higher-order functions (HOFs), accompanied by XLink/XPointer constructs, in facilitating data extraction from native data repositories. In figure 4, an example of a *dfdl_ext:filter* function is illustrated. This higher-order function takes a *predicate* function and list, and returns to the *money laundering application* the list of elements that satisfy the predicate. Note another HOF construct is given in figure 3. To facilitate the data extraction, each DFDL processor is pre-complied, and the *function definition* is passed to the DFDL parser's runtime environment by way of a XLink arc traversal. As a result, the analyst is able to ignore the details of the data model and URI-based location of data fragment, and implement functional queries based on the conceptual modeling of the money laundering domain.

VIII. Compiler Specification for Data Description Language extended with HyperLink Structures and Higher-Order Functions (HOF)

This section addresses the initial specification of a DFDL compiler using CRSX to perform syntactic analysis, semantic analysis and transformation of a DFDL instance into a higher-order abstract syntax (HOAS) intermediate language. As a mathematical rewriting method, CRSX is used to formalize a *stepwise* transformation and evaluation of the concrete DFDL language into a highly optimized parsing application or raw data writer. By way of a recursive tree traversal over a DFDL schema instance, the CRSX-based compiler steps through each of the following compilation phases:

A. Syntax to CFG Production Rule for DFDL

First, the names for all components of the DFDL schema language are specified. In this case, a *component* is anything that can be defined or declared in the DFDL vocabulary (e.g., an element, a simple type or a complex type). As in figure 3, more than four hundred components belonging to the DFDL namespace were specified. Second, for each specified name in the language, explicit transformation mappings were made from the DFDL schema syntax onto context-free grammar (CFG) structures.

A DFDL schema fragment is illustrated in figure 3, by the xs:complexType declaration example. This particular code fragment is related to defining a complexType element that contains other elements such as choice and sequence. As the CFG for complexType is used to guide the parse into each fragment of the DFDL schema syntax, essentially an invocation of a unit of instruction is sent to the DFDL compiler to be executed. The XML markup in the DFDL schema express structure. The process of parsing a DFDL schema identifies elements and attributes, and creates an abstract image (i.e., the DFDL data model) that corresponds to the DFDL schema structure.

Consider, for example in figure 4, one of the DFDL element declarations. On "line 9", it declares to be an instance of a ’dfdl:element’ element type. The DFDL element must comply with the structure and attribute constraints stated by the element type in order to qualify as an instance. In this case, membership of an instance of an DFDL element or attribute to a type is determined by validation of a DFDL processor that is tasked to accept or reject DFDL instances as well as data mapped to those instances. In the case of the syntax for XLink, XPointer and higher-order functions (ref: figure 3) used in the extended DFDL schema, CFG production rules are also prescribed in a similar manner.

Further, a parallel exits between the DFDL schema and the data file which validated against the DFDL schema. Use of a CFG is the approach taken by the compiler for providing

```
<?xml version="1.0" encoding="UTF-8"?>
<linkbase xmlns:xlink="http://www.w3.org/1999/xlink" xlink:linkbase="http://www.w3.org/1999/xlink/properties/linkbase">
  <link xlink:type="extended" xlink:title="moneyLaunderLinkbase">
    <!-- Linkbase loads on extraction request. -->
    <basesloaded>
    <startrsrc xlink:label="filter_spec" xlink:href="/local/filter_spec.xml#params" />
    <linkbase xlink:label="linkbase" xlink:href="/local/linkbase.xml" />
    <load xlink:from="spec" xlink:to="linkbase" actuate="onRequest" />
    </basesloaded>

    <!-- Arcs between linkbase and DFDL-data stores. -->
    <invokeStoreArc xlink:type="arc" xlink:arcrole="linkbase" xlink:from="linkbase" xlink:from="PolicyAccountRecord"/>
    <invokeStoreArc xlink:type="arc" xlink:arcrole="linkbase" xlink:from="linkbase" xlink:from="CancelRequestRecord"/>
    <invokeStoreArc xlink:type="arc" xlink:arcrole="linkbase" xlink:from="linkbase" xlink:from="RefundRequestRecord"/>

    <!-- Locator elements. -->
    <loc xlink:type="locator" xlink:label="PolicyAccountRecord" xlink:href="http://tmp1.linuz.org/policyAccountSchema.dfdl"/>
    <loc xlink:type="locator" xlink:label="PolicyAccountIdentifier" xlink:href="http://tmp1.linuz.orglinuz1/policyAccountSchema.dfdl#xpointer(////policyAccountIdentifier[@xs:string=value])"/>
    <loc xlink:type="locator" xlink:label="PolicyStartDate" xlink:href="http://tmp1.linuz.org/policyAccountSchema.dfdl#xpointer(////policyStartDate[@xs:date=value])"/>
    <loc xlink:type="locator" xlink:label="PolicyHolder" xlink:href="http://tmp1.linuz.org/policyAccountSchema.dfdl#xpointer(////policyHolder[@xs:string=value])"/>
    <loc xlink:type="locator" xlink:label="PolicyInsurer" xlink:href="http://tmp1.linux.org/policyAccountSchema.dfdl#xpointer(////policyInsurer[@xs:string=value])"/>
    <loc xlink:type="locator" xlink:label="PayerName" xlink:href="http://tmp1.linuz.org/policyAccountSchema.dfdl#xpointer(////payerName[@xs:string=value])"/>
    <loc xlink:type="locator" xlink:label="PayerCurrency" xlink:href="http://tmp1.linuz.org/policyAccountSchema.dfdl#xpointer(////payerCurrency[@xs:string=value])"/>
    <loc xlink:type="locator" xlink:label="PremiumAmount" xlink:href="http://tmp1.linuz.org/policyAccountSchema.dfdl#xpointer(////premiumAmount[@xs:string=value])"/>
    ...

    <!-- Relationship between policy account, cancel request and refund request identifiers. -->
    <invokeIdArc xlink:type="arc" xlink:arcrole="owl:sameAs" xlink:from="PolicyAccountIdentifier" xlink:to="CancelRequestIdentifier"/>
    <invokeIdArc xlink:type="arc" xlink:arcrole="owl:sameAs" xlink:from="PolicyAccountIdentifier" xlink:to="RefundRequestIdentifier"/>
    <invokeIdArc xlink:type="arc" xlink:arcrole="owl:sameAs" xlink:from="CancelRequestIdentifier" xlink:to="PolicyAccountIdentifier"/>
    <invokeIdArc xlink:type="arc" xlink:arcrole="owl:sameAs" xlink:from="CancelRequestIdentifier" xlink:to="RefundRequestIdentifier"/>
    <invokIdArc xlink:type="arc" xlink:arcrole="owl:sameAs" xlink:from="RefundRequestIdentifier" xlink:to="PolicyAccountIdentifier"/>
    <invokeIdArc xlink:type="arc" xlink:arcrole="owl:sameAs" xlink:from="RefundRequestIdentifier" xlink:to="CancelRequestIdentifier"/>

    <!-- Relationship between policy account record and its parts. -->
    <invokeParArc xlink:type="arc" xlink:arcrole="owl:hasPart" xlink:from="PolicyAccountRecord" xlink:to= "PolicyAccountIdentifier" />
    <invokeParArc xlink:type="arc" xlink:arcrole="owl:hasPart" xlink:from="PolicyAccountRecord" xlink:to= "PolicyStartDate" />
    <invokeParArc xlink:type="arc" xlink:arcrole="owl:hasPart" xlink:from="PolicyAccountRecord" xlink:to= "PolicyHolder" />
    <invokeParArc xlink:type="arc" xlink:arcrole="owl:hasPart" xlink:from="PolicyAccountRecord" xlink:to= "PolicyInsurer" />
    <invokeParArc xlink:type="arc" xlink:arcrole="owl:hasPart" xlink:from="PolicyAccountRecord" xlink:to= "PayerCurrency" />
    <invokeParArc xlink:type="arc" xlink:arcrole="owl:hasPart" xlink:from="PolicyAccountRecord" xlink:to= "PayerName" />
    <invokeParArc xlink:type="arc" xlink:arcrole="owl:hasPart" xlink:from="PolicyAccountRecord" xlink:to= "PremiumAmount" />
    ...
   </link>
</linkbase>
```

Fig. 7. The money laundering linkbase.

```
(a.1) Input "policy account record" data:

PLCYACC/741032-1071//
DATE/2013-09-28//
PLCYHLD/Allegier, Cox & Associates, Inc.//
INSUR/ALI Corp.//
PAYER/Grupo Palermo S.A.//
PAYCUR/Peso (ARG)//
PRMAMT/42004.98//

(a.2) DFDL generated XML model:

<policyAccountRecord>
    <policyAccountIdentifier>741032-1071</
            policyAccountIdentifier>
    <policyStartDate>2013-09-28</policyStartDate>
    <policyHolder>Allegier, Cox & Associates Inc.
           </policyHolder>
    <policyInsurer>ALI Corp.</policyInsurer>
    <payerName>Grupo Palermo S.A.</payerName>
    <payerCurrency>Peso (Argentine)</
            payerCurrency>
    <premiumAmount>42004.98</premiumAmount>
</policyAccountRecord>

(b.1) Input "credit request record" data:

[[[[[[[CRDREQ%]741032-1071%]
CRDATE%]2013-11-02%]
PAYEE%]Allegier, Cox & Associates, Inc.%]
PAYCUR%]USD%]
CRDAMT%]5000.00%]

(b.2) DFDL generated XML model:

<creditRequestRecord>
    <creditRequestIdentifier>741032-1071</
            creditRequestIdentifier>
    <creditRequestDate>2013-11-02</
            creditRequestDate>
    <payeeName>Allegier, Cox & Associates, Inc.</
            payeeName>
    <payeeCurrency>USD</payeeCurrency>
    <creditAmount>5000.00</creditAmount>
</creditRequestRecord>
```

Fig. 2. *Policy Account* and *Credit Request* records. A DFDL *parser* accepts raw data (e.g., in (b.1)) and generates a DFDL data model (in (b.2)). Symmetrically, a DFDL *unparser* uses a DFDL data model generate equivalent raw data.

syntactic checking. Ultimately, the aggregate of four hundred or so CFG production rules will partition any DFDL schema into a set of components, where each component can match against an unique fragment of a DFDL schema. The CFG has been designed to ensure that any DFDL schema be reduced to its normal form in order to provide a specific name for each component of the DFDL schema specification.

B. CFG Production Rule to HOAS for DFDL

Next, rules for transformation of the CFG into the HOAS intermediate language are prescribed in the DFDL compiler implementation. Note in figures 3 and 4, a HOAS *constructor* name is shown for each provided CFG. A HOAS representation is the equivalent to an abstract syntax tree (AST), and it serves as the intermediate representation for further transformation and optimizations of a DFDL schema instance. As illustrated in the HOAS column of figure 3, a HOAS surmounts the difficulty of having to define name binding constructs in the abstract syntax [9]. For example, *XsComponentType* is a syntactic category and *ComplexType*, which is a name of a type that has membership to that category, is bound using the [ ] syntax. All the DFDL types

Concept	Example
Classes	propertyAccountRecord, cancelRequestRecord and creditRequestRecord (ref: figs. 5 and 8)
Instances	An instance of a propertyAccountRecord is one bearing "741032-1071" as the policyAccountIdentifier (ref: fig 2, a.1).
Relations: hasPart, sameAs	The three properties, policyAccountIdentifier, cancelRequestIdentifier, and creditRequestIdentifier are equivalent (sameAs) (ref. figs. 8 and 9).
Properties	policyAccountIdentifier, policyStartDate, policyHolder, policyInsurer are properties of a policyAccountRecord (ref: figs. 5, 8 and 9).
Values	"USD" and "5000.00" are the values of payeeCurrency and premiumAmount respectively for a particular instance of a creditRequestRecord (ref: fig. 2, b.2).
Rules	The three properties, policyAccountIdentifier, cancelRequestIdentifier, and creditRequestIdentifier are equivalent (sameAs) if they evaluate to the same value, for example, "741032-1071".

Fig. 5. This table gives an explanation of the anti-money laundering conceptual ontology and illustrates where these concepts are defined within XLink-extended DFDL schemas and linkbase. [2]

Attribute	Value	Description
xlink:type	extended	Parent element, which defines a complex link in which multiple links can be combined based on other attributes.
	resource	Child element of extended-Type element, which provides a local resource to associate with the link.
	locator	Child element of extended-Type element, which specifies the location of a remote resource associated with the link.
	arc	Child element of extended-Type element, which define traversal rules between the link's associated resources.
xlink:label	NCName	Traversal attribute of extended-, resource-Type elements, which provides a reference (of itself) to arc-Type in composing a traversal arc.
xlink:from, xlink:to	NCName	Traversal attributes of arc-Type element, which define the source and target resources of the arc link.
xlink:href	URI	Attribute of locator-Type element, which provides the data that helps an XLink application to locate a remote resource .
xlink:role	URI	Semantic attribute of extended-, resource-Type elements, which indicates a property of the resourcein a computer readable-form.
xlink:arcrole	URI, linkbase	Semantic attribute of arc-Type element, which coincides with the [RDF] view of a property, where the role can be understood as HAS relationship between the starting-resource and the ending-resource.
#xpointer		Creates XPointer fragment links with syntax: #xpointer(id("<value>"))

Fig. 8. XLink elements and attributes used in the anti-money laundering application. [4]

```
TERM ::=(
  Let[ VALUE, TYPE, x::VALUE . TERM ];
  Lam[ VALUE, TYPE, x::VALUE . TERM ];
  Context[ ];
  Element[KIND, $List[ATTRIBUTE], $List[DFDL_PROPERTY], $List[XLP_ATTRIBUTE], TERM];
  Pair[ TERM, TERM ];
  Nil;
  T;
  T-Attribute
  T-BuildSchema
  T-BuildElement
  XML-Visit[ XLink-XPointer ]
);
```

Fig. 9. Consider our top level terms for the DFDL CRSX system after normalization. The terms are written in the form of a higher-order abstract syntax.

(including the new extensions) are derived from *syntactic sorts* or *syntactic categories*, which are normalized into the top level terms of the DFDL HOAS shown in figure 9. The objective is to convert all CFG derivations into the syntactic sorts. For example, in the normalization process, the "XsComponentType[ComplexType]" would be transformed to the term: "Element[XsComponentType[ComplexType]], ...]".

C. HOAS to CRSX Rewrite Rule for DFDL

CRSX *rewrite rules* are specified to address the semantic and optimization transformation and evaluation of the DFDL compiler. An example *rewrite rule* is given (in figure 10) that defines explicit scoping of XS COMPONENT ATTRIBUTE(s) in the DFDL specification. The meaning of the *rewrite rule* syntax is provided in [10]:

(a) A *rewrite rule* takes the form:

$$name[options] : pattern \rightarrow contraction$$

, where name should be a *constructor* and the pattern and contraction should be *terms*; where

(b) *XsComponent-Attribute* and *Copy* are constructors, which take an optional ordered or positional *parameter* list in immediately following [ ]s, where each parameter is itself a *term*, and called a *subterm*, uncapitalized words (e.g. x and foo) denote *variables*; and where

(c) a *Lambda-construction* with a single subterm binds the variable x (before the .) and contains a single construction with two subterms that both are occurrences of x.

In CRSX, we model this as Let[E1, x.E2], i.e. let all *occurrences* of x in the body of function E2 be replaced or substituted by x, where x := E1. This allows explicit scoping. The entire compiler is specified as rule system is written as a sequence of rules each terminated by ; semicolon;

IX. Conclusion and Future Work

In this paper, an approach is given for a lightweight-capability that supports metadata-based discovery and extraction of *informational* fragments from raw data stores without having to alter the information's native data format.

```
XsComponent-Attribute[Copy[#QName]]
:
{#Env; #QName: ComponentAttribute[#kind]}
XML-Attribute[ #prefix, #QName, #Value, ok.#Continuation[ok]]
->
{#Env}
 Let[ #Value, a.{#Env} AddXsAssoc[#prefix, #QName, a, ok.#Continuation[ok]]
;
```

Fig. 10. Illustrates a CRSX *rewrite rule* that defines explicit scoping of XSD attributes. CRSX *rewrite rules* are an equivalent form of a programming language's *operational semantics*. In this case, we define the operational semantics from the perspective of compilation.

This approach offers an advantage over popular data extraction schemes such as Apache Hadoop platform that require the conversion of data into a *prescriptive* data format. The approach extends an existing *data description language* with linking/-pointing constructs and higher-order functions. An overview of the DFDL compilation is provided using concepts from programming language design and formal rewriting systems. The future work includes: specifying the transformation and evaluation of the DFDL/XLink/HOF specification into parser combinator form; investigating the operational semantics of the higher-order function (HOF) and linking abstractions in order to optimize distributed data extraction; and generating comparative performance metrics.

Acknowledgment

The authors would like to thank Dr. Kristoffer Rose for his valuable guidance on Higher-order Rewriting for Compiler Specifications and the Combinatory Reduction System with Extensions.

References

[1] O. D. WG, S. M. Hanson, and A. W. Powell, "Data format description language (dfdl) v1. 0 specification."

[2] K. H. Rose, "Crsx–an open source platform for experiments with higher order rewriting," *HOR 2007*, p. 31, 2007.

[3] K. Fisher and R. Gruber, "Pads: a domain-specific language for processing ad hoc data," in *ACM Sigplan Notices*, vol. 40, no. 6. ACM, 2005, pp. 295–304.

[4] M. Sporny, G. Kellogg, and M. Lanthaler, "Json-ld 1.0-a json-based serialization for linked data," *W3C Working Draft*, 2013.

[5] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[6] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A distributed storage system for structured data," *ACM Transactions on Computer Systems (TOCS)*, vol. 26, no. 2, p. 4, 2008.

[7] I. A. of Insurance Supervisors, *Examples of Money Laundering and Suspicious Transactions Involving Insurance*. International Association of Insurance Supervisors, 2004. [Online]. Available: http://books.google.com/books?id=bSvoHAAACAAJ

[8] S. DeRose, E. Maler, D. Orchard, and N. Walsh, "Xml linking language (xlink) version 1.1, w3c recommendation 06 may 2010," 2010.

[9] F. Pfenning and C. Elliot, "Higher-order abstract syntax," in *ACM SIGPLAN Notices*, vol. 23, no. 7. ACM, 1988, pp. 199–208.

[10] K. H. Rose, "Crsx-combinatory reduction systems with extensions," in *LIPIcs-Leibniz International Proceedings in Informatics*, vol. 10. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2011.

Security Requirements Analysis of ADS-B Networks

Thabet Kacem, Duminda Wijesekera, Paulo Costa
Center of Excellence in C4I
George Mason University
Fairfax, Virginia
[tkacem, dwijesek, pcosta]@gmu.edu

Alexandre Barreto
Instituto de Controle do Espaço Aéreo
Centro Tecnológico da Aeronáutica
São José dos Campos, SP - Brazil
barretoabb@icea.gov.br

Abstract— **Due to their many advantages over their hardware-based counterparts, Software Defined Radios are becoming the new paradigm for radio and radar applications. In particular, Automatic Dependent Surveillance-Broadcast (ADS-B) is an emerging software defined radar technology, which has been already deployed in Europe and Australia. Deployment in the US is underway as part of the Next Generation Transportation Systems (NextGen). In spite of its several benefits, this technology has been widely criticized for being designed without security in mind, making it vulnerable to numerous attacks. Most approaches addressing this issue fail to adopt a holistic viewpoint, focusing only on part of the problem. In this paper, we propose a methodology that uses semantic technologies to address the security requirements definition from a systemic perspective. More specifically, knowledge engineering focused on misuse scenarios is applied for building customized resilient software defined radar applications, as well as classifying cyber attack severity according to measurable security metrics. We showcase our ideas using an ADS-B-related scenario developed to evaluate our research.**

Keywords— Ontologies, Misuse case, Cybersecurity, ADS-B

I. Introduction

Since first proposed in 1991 by Joe Mitola [1], the concept of Software Defined Radio (SDR) has received considerable research interest. The idea of migrating hardware-based radio functionality to software and, among other benefits, dynamically optimize the spectrum use is compelling. Not surprisingly, the concept is now applied to applications whose focus spans from cognitive radios to radar applications.

In particular, *Air Traffic Control* (ATC) systems research became a natural area for SDRs, due to a pressing need for modernizing its current standards, most stemming from the 1970s. In this context, *Automatic Dependent Surveillance-Broadcast* (ADS-B) has emerged as the leading technology for radar surveillance, and has been already deployed in Europe, Canada and Australia. The U. S. Federal Aviation Administration plans to have it deployed by 2020 as part of NextGen [2]. Most aircraft manufacturers are already equipping their newest models with ADS-B, which is present in aircraft such as the Boeing 777 and the Airbus A380.

In spite of its success, ADS-B has several critics. Strohmeier et al. [3] point out the huge lack of security caused by the fact that ADS-B sends its packets in clear text, making it vulnerable to attacks that target the confidentiality, integrity, availability and non-repudiation properties of the data. This concern is consistent with research done by Costin and Francillon [4] and Schäfer et al. [5], who described the possibility of eavesdropping, man-in-the-middle and denial of service attacks in simulated environments.

Unfortunately, most research efforts aimed at detecting and mitigating ADS-B vulnerabilities lack a systems engineering methodology, therefore failing to address the problem from a holistic perspective. For instance, many lack a comprehensive approach to perform attack analysis and mitigation, and assess their impact on applications of ADS-B technology, which we propose in this paper. Conversely, system engineering techniques such as *use cases* and *interaction diagrams* have been widely used in other domains to model the system's behavior and its interaction with users, which is done from the early designing steps in the system lifecycle.

Misuse cases [6] extend the concept of use case development to model potential undesirable behaviors. The technique has been gaining popularity in recent years as a means to enhance system security, by modeling undesirable behaviors, ensuring these are addressed during systems design. Misuse cases represent the threats to a system at a high level perspective, while the step-by-step details are represented using *mal-activity diagrams*. The latter is key for devising ways of thwarting attacks, but both are essential for designing resilient systems.

Another important technology for designing systems is *Ontology engineering*. Ontologies represent knowledge within a specific domain by formally describing its key concepts and the relationships among them. They allow for automated knowledge management and discovery via logical inferences and have been applied to a variety of applications, such as health care and artificial intelligence. Yet, there has been a surprising lack of research in the ontology community for designing secure SDR applications, and only a few have been proposing to leverage ontologies in this area (e.g. [7]).

Our work bridges this gap by proposing a new methodology for building resilient SDR applications that relies on ontologies. We leverage their reasoning capabilities to automate the modeling of use cases, misuse cases, mal-activity diagrams, mitigation case diagrams and mitigation activity diagrams, all within the design phase of the radar application in question. We present the approach in this paper, and contextualize our ideas using an ADS-B ATC scenario.

Our methodology brings three main contributions to the design of secure ADS-B systems. First, we applied semantic technologies in support to security and requirements modeling, formalizing knowledge relevant to SDR systems for building resilient radar networks. To the best of our knowledge, this is the first approach to do so. Other research efforts that leveraged security ontologies either focused on security in general, such as [8], or on a specific domain, such as [9]. The work in this paper formalizes the knowledge of secure ADS-B systems in a way that can be extended to other SDR applications.

The second major contribution of this paper is the application of inferential reasoning to enhance security-related design activities. Examples include automated verification of whether the mal-activity and mitigation activity diagrams are consistent with misuse case and mitigation case diagrams respectively, and checking whether the mitigation techniques can effectively thwart the potential attacks. In this initial work, we used Protégé [10] to develop our ontology and the Pellet reasoner [11] to support the automated verification.

The third major contribution of our paper is the development of measurable security metrics to classify the detected attacks according to a taxonomy that we also define in this paper. We use the work in [12] as a reference when developing the metrics we defined for ADS-B applications.

The rest of the paper is organized as follows. Section II presents background information about ADS-B and enumerates some of the security issues discussed in the literature. Section III describes our methodology. Section IV illustrates the use of our methodology by presenting an application scenario. In Section V, we show how to classify the detected attacks using ontological rules and security metrics. Section VI describes related work in securing ADS-B applications, misuse cases, and mal-activities in security-related operations. Section VII has our conclusions.

II. Background

One of the main contributions of ADS-B to ATC is its ability to provide better coverage, flexibility, cost-effectiveness, and simplicity than traditional radar. Further, ADS-B can either extend radar coverage or provide a similar service in locations without radar coverage - such as oceanic routes. It also reduces cockpit activity, since pilots would not need to provide constant updates. The costs involved in deploying and operating an ADS-B station are much lower than those observed in traditional radar stations [13].

The ADS-B protocol has two modes of operation: ***ADS-B Out*** and ***ADS-B In***. The first broadcasts aircraft position along with other data over the 1090MHz frequency band for commercial flights and 948MHz band for general aviation. ADS-B packets are encapsulated in *Mode S Extended Squitter* frames consisting of an 8 bits preamble used for synchronization and a 56 or 112 bits data block containing the ADS-B data. It is modulated via *Pulse Position Modulation* (PPM) at 1 Mbit per second rate. *ADS-B In* receives broadcasts from nearby aircraft. This feature is mostly used by ATC services so its deployment is not mandatory to aircraft.

ADS-B presents considerable advances when compared to *Primary Surveillance Radar* (PSR), which determines the approximate aircraft position by measuring the time a reflected pulse takes to reach back to its originating radar antenna. Because the emitted pulse is many orders of magnitude greater than the incoming reflected pulses, radar circuitry is extremely complex. ADS-B also has an advantage over *Secondary Surveillance Radars* (SSR), which relies on aircraft-borne transponders to transmit their positions. Unlike ADS-B, SSR must rely on cooperation by pilots and – mostly for that reason - its operation tends to be error-prone.

In spite of these advantages, ADS-B has its own share of limitations due to its vulnerability to cyber attacks. Several publications on ADS-B security (e.g. [4], [5] and [14]) used a simulated environment to demonstrate various types of attacks targeting this technology, mostly using low cost equipment. The primary source of vulnerability is that data is sent in clear text, without authentication or encryption. Some of the ADS-B attacks demonstrated in simulated environments are:

- **Eavesdropping:** performed with low cost radio devices operating at 1090 MHz combined with an open source implementation of ADS-B receiver. Basically, one can eavesdrop on all air traffic within the range of the radio device. Although eavesdropping is technically not an attack by itself, it is a prerequisite step for many others.
- **Injection Attacks:** performed by an attacker that emits ADS-B messages referencing a fake aircraft (i.e. "injects" a fake aircraft) that interacts with the trajectory of a real aircraft, forcing its pilot and the ATC services to adopt unintended actions to avoid collisions. These attacks usually rely on a preceding eavesdropping phase for capturing the required parameters.
- **Denial of Service:** these are basically a "brute force" version of injection attacks, if less elaborate. The idea is to insert a large number of fake aircrafts to the ATC's screen, causing a denial of service. Air traffic controllers will not be able to distinguish fake aircraft from real ones, or to prevent system crashes due to the heavy load.
- **Man in the middle:** these can be variations of the above, but with a person in the control of the attack. It is possible for an attacker to intercept live traffic, store ADS-B packets, modify them and retransmit the tampered ones back to create confusion in air traffic control operations.

III. Methodology

The main goal of our work is to help the software architect in designing the core system components with security as a first class citizen, instead of an afterthought. A key concept is our reliance on ontologies to provide the ADS-B system designer with an automated way of testing the security features in a cohesive fashion. We adopted Protégé [10] in this research due to its popularity and built-in reasoners, such as Pellet [11] - which we use to verify the correctness of the attack mitigation techniques. Figure 1 shows a high-level view of our methodology, and highlights the input it requires from the systems engineer. More specifically :

- Use case diagrams: system functionalities.
- Misuse case diagrams: undesired functionalities.

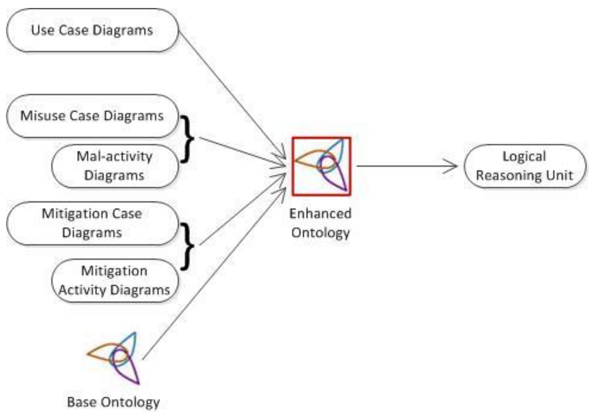


Figure 1: High level view of the methodology

- Mal-activity diagrams: sequence of actions refining a Misuse case.
- Mitigation case diagrams: counteractive functionalities that detect or mitigate undesired functionalities.
- Mitigation activity diagrams: sequence of actions that refine a Mitigating use case.
- Base ontology: Models classes, their relationships, and properties of the SDR domain.

Figure 2 shows the main concepts we have included in the base ontology. Our methodology precisely defines the meaning of "security" by specifying security in terms of desired and undesired system behavior. The proposed design process involves supporting the system designer to model the required and undesirable system functionalities using the classes, inter-class and intra-class relationships, and properties of the base ontology to produce the input listed in Figure 1. Logical reasoning is used in the process to ensure that the design entails the desired and undesired system properties, empowering the design team with an automated verification of the fact that their design is compliant with the design's security objectives (i.e. design by contract). The process outcomes can also be used as formal, accountable artifacts that can be independently verified.

IV. EVALUATION

To evaluate if our ideas would result in a more secure ADS-B network, we have designed an ATC scenario and developed an ADS-B application for this scenario using the proposed methodology. Our scenario includes a network topology consisting of the following components:

- Helicopters: the scenario focus on a helicopter operation that is supported by an ADS-B network.
- ATC Center: one main ATC Terminal receives aircraft track information from a radar and an ADS-B server to provide navigation instructions to the helicopters.
- ATC server: receives location updates from the remotely connected telecommunication sites and ADS-B stations. It stores the updates in a database and evaluates pre-defined security constraints, such as separation between helicopters.
- Telecommunication Sites: the scenario includes three (named T1, T2 and T3), which convey location updates to the ATC server and broadcast navigation instructions to helicopters using ADS-B stations.
- ADS-B Stations: each of the five stations (s1 to s5) receives ADS-B packets from helicopters, broadcasts these to the telecommunication site it is connected to, and forwards navigation instructions to the pilots.
- Communication Links: transmit data between the ATC server, ADS- B stations and telecommunication sites.

Our scenario leverages work such as Magazu [14] shows that attacking ADS-B networks can be relatively simple and inexpensive by purchasing a basic radio device (even a computer dongle) and using open source software such as GNU Radio [15] and Gr-Air-Modes [16] as an ADS-B receiver. In the scenario, the attacker can:

- **Tamper position:** The attacker receives location updates from a legitimate helicopter using an ADS-B receiver placed within the transmission range. Then, modifies ADS-B packets by either altering the hexadecimal content or by inserting GPS coordinates into the packet that may be inconsistent with the logical flight path.
- **Create a ghost helicopter:** The attacker introduces a new helicopter in the range of an ADS-B station so that it gets propagated to the ATC center, and consequently alters flight paths of legitimate helicopters. That is, if the fake trajectory interferes with the real aircraft, this will force active interference by the flight controllers.
- **Flood an ADS-B station:** The attacker overwhelms an ADS-B station with fake packets, affecting the control of helicopters within range of that station. That is, a

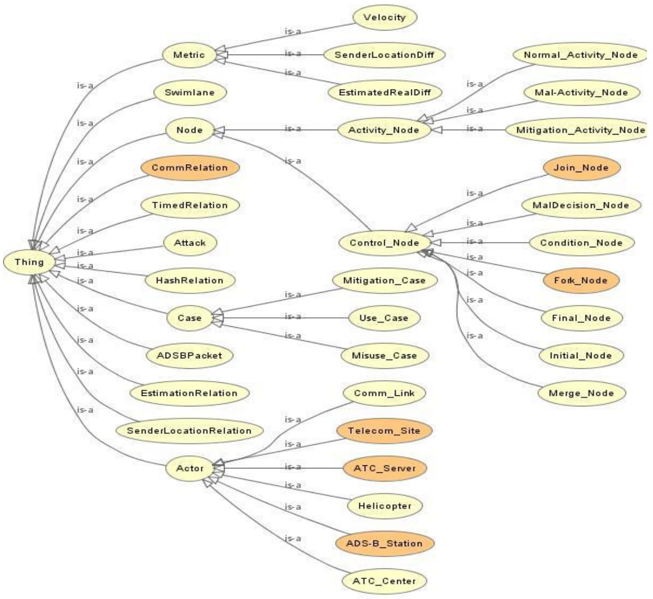


Figure 2: Base Ontology

Denial-of-Service attack.

- **Flood ATC/RADAR:** similar to the previous attack but this is done at a larger scale to overwhelm many or all ADS-B stations. If successful, this would adversely affect regional air traffic.

The following mitigations are viable against these attacks:

- **Check Hash:** Embed a hash of the ADS-B payload in the packet to preserve data integrity using pre-shared hashing metadata.
- **Rate Limiting:** Rate control the packets received from communication links of the ADS-B stations.

The core functionalities of the application are represented with use cases. The attacks to be prevented during the design phase are represented using misuse cases, and counteraction techniques are represented using mitigation cases. Taken together, these represent the high-level security objectives of the system.

To achieve security design objectives, our methodology requires more detail from the system architect, who has to define diagrams conveying the activities, mal-activities, and mitigation activities – all consistent with contemporary design activities for large-scale resilient systems. Figure 3 illustrates the combined view of these diagrams. In the figure, every lane is annotated with a name of an actor and the actions. Black ovals indicate mal-activities while white ovals indicate normal or mitigation activities. To facilitate understanding for the methodology, we now provide an overview of each lane:

- Helicopter lane: the black-filled circle designate the start of the “Broadcast location” usage scenario. Ovals “Get self location” and “Broadcast location” designate the two activities that are responsible, respectively, for getting the location of the helicopter and sending it via ADS-B Out. The black rectangle indicates a fork node. It models how location data is broadcasted to all nearby helicopters.
- Attacker lane: the three back ovals show how the misuse case “Tamper position” works. Oval “Receive location” indicates that the attacker received the location update from the legitimate helicopter. Oval “Tamper location” describes how the attacker crafts fake location inside the ADS-B packet while oval “Send fake location” broadcasts the altered packet back to the nearby ADS-B stations.
- ADS-B Station 2 lane: the black rectangle indicates a join node showing how the ADS-B station receives location updates from the helicopter and the attacker. The two ovals “Receive location” and “Send location” in this lane are two activities as part of “Replay Data” use case.
- Telecom Site 1 lane: the two ovals “Receive location” and “Send location” are also part of “Replay Data” use case and show how the ADS-B packets are replayed through the telecom sites.
- Comm Link 1 lane: oval “Transmit” indicates how the data inside the packet is physically transmitted. This activity is part of “Transmit data” use case.
- ATC Server: oval “Receive” designates that the ADS-B packets are received. However, oval “Check Hash” represents a mitigation activity as part of “Check Hash” mitigation case. It indicates that the ATC server checks the received hash against the hash it computes based on the payload of the received packet. The diamond indicates a decision node. Based on the outcome of the computation of the above described condition, the ATC server directs the flow of the whole scenario accordingly. If the result is a mismatch, then it connects to the oval “Discard” which is a normal activity indicating that the ATC server would just ignore the packet before ending the scenario by connecting to the double-edged black circle.
- ATC Center lane: if the result of the previous decision is a match, the oval “Display air traffic” will be connected. This oval is part of “Display air traffic” use case”. Similarly, the scenario would end at this point by connecting to the double-edged black circle.

All the elements of Figure 3 can be mapped to the base ontology classes where, each lane is an individual of the Swimlane class and every label has the actor’s name. However, this mapping depends on the characteristics of each sub-class of Actor. More specifically:

- Helicopter and Attacker: mapped to the Helicopter class.
- ADS-B station 2: mapped to the ADS-B_Station class.
- The black-filled circle: mapped to an individual of the Initial_Node class
- Double-edged black circle: mapped to an individual of the Final_Node class.
- Black rectangles: can be mapped to either the Join_Node class or the Fork_Node class, depending on the incoming and outgoing arrows. This is modelled by ontological restrictions linking each member of this class to the number of instance of the Node class connected to it.
- White ovals: indicate a normal activity and are

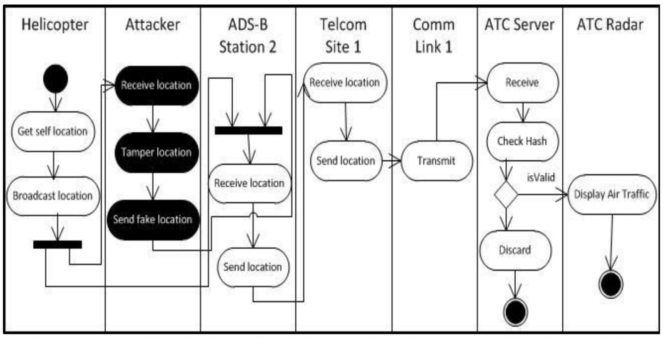


Figure 3: Combined view of activity, mal-activity and mitigation activity diagrams

considered individuals of the Normal_Activity_Node, while those indicating a mitigation activity are considered individuals of the Mitigation_Activity_Node.

- Black ovals: are individuals of the Mal-Activity_Node.

Arrows connecting the elements described above are mapped to object properties that relate two instances of two different classes. In our work, this is done using (Protégé) ontology rules, previously known as Semantic Web Rule Language (SWRL) rules [17]. Each rule implies the *consequent* (right hand side, a.k.a. head) from the antecedent (left hand side, a.k.a. body).

Let π be the statement of Theorem 1, described in Listing 1. It shows a rule that models the fact that “tamper location” misuse case “threatens” the “monitor air traffic” use case. The rule is part of the “threatens” use case / misuse case relationship in the scenario where every helicopter sends an ADS-B packet containing the required information.

Each packet has a location defined as {latitude, longitude, altitude}. When two packets sent from two different helicopters reach the ATC Server, the server compares their timestamps and their locations. If the timestamps are the same and the differences in the received longitudes, latitudes, and altitudes are greater than the predefined ε, then the reasoner will infer the “threatens” object property. The Pellet reasoner then gathers the data and object properties of the individuals concerned by the defined rule, and tries to infer the head - in this case the “threatens” object property. If it succeeds in doing “threatens” will appear as highlighted and we can get the corresponding explanation.

Let ψ be the statement of Theorem 2 provided in Listing 2. It shows the rule proving that the mitigation case succeeds in thwarting the previously detected misuse case that threatens a given use case of the system. It also tags the actor in question as malicious, and associates the attack with its swimlane (cf. Figure 3). Therefore tagging the associated object properties as the names of the mal-activities associated with the attack.

```
ADSBPacket(?p1), ADSBPacket(?p2),
 ATC_Server(?a), Helicopter(?h1), Helicopter(?h2),
Misuse_Case(Misuse_Case_1),
TimedRelation(?tr1), TimedRelation(?tr2),
Use_Case(Use_Case_3),
broadcastADSBPacket(?h1, ?p1),
broadcastADSBPacket(?h2, ?p2),
genADSBPacket(?h1, ?p1), genADSBPacket(?h2,
?p2), receives_updates(?h2, ?p1),
receives_updates1(?a, ?tr1), receives_updates1(?a,
?tr2), receives_updates2(?tr1, ?p1),
receives_updates2(?tr2, ?p2), altitude(?p1, ?alt1),
altitude(?p2, ?alt2), icao(?p1, ?id1), icao(?p2, ?id2),
latitude(?p1, ?lat1), latitude(?p2, ?lat2),
longitude(?p1, ?long1), longitude(?p2, ?long2),
time_value(?tr1, ?t1), time_value(?tr2, ?t2),
subtract(?dalt, ?alt1, ?alt2), subtract(?dlat, ?lat1,
?lat2), subtract(?dlong, ?long1, ?long2), abs(?adalt,
?dalt), abs(?adlat, ?dlat), abs(?adlong,  ?dlong),
equal(?t1, ?t2), greaterThanOrEqual(?adalt, 50),
greaterThanOrEqual(?adlat, 0.05),
greaterThanOrEqual(?adlong, 0.02), →
threatens(Misuse_Case_1, Use_Case_3)
```

Listing 1

```
ADSBPacket(?p1), ATC_Server(?a),
HashRelation(?pi1), Helicopter(?h1), Mal-
Activity_Node(Mal-Activity_1), Mal-
Activity_Node(Mal-Activity_2), Mal-
Activity_Node(Mal-Activity_3),
Misuse_Case(Misuse_Case_1),
Mitigation_Case(Mitigation_Case_1),
Swimlane(?s1), TimedRelation(?tr1),
comprise(Misuse_Case_1, Mal-Activity_1),
comprise(Misuse_Case_1, Mal-Activity_2),
comprise(Misuse_Case_1, Mal-Activity_3),
computes_hash1(?a, ?pi1), computes_hash2(?pi1,
 ?p1), genADSBPacket(?h1, ?p1), occupies(?h1,
?s1), receives_updates1(?a, ?tr1),
receives_updates2(?tr1, ?p1),
computed_hash(?pi1, ?cHash1), hash(?p1,
?hash1), notEqual(?hash1, ?cHash1) →
Attack(TamperPosition),
isAssociatedWith(TamperPosition, ?s1),
mitigates(Mitigation_Case_1, Misuse_Case_1),
 participates_in(?h1, Misuse_Case_1),
associated_obj_prop(Mal-Activity_1,
"receives_updates"^^string),
associated_obj_prop(Mal-Activity_2,
"genADSBPacket"^^string),
associated_obj_prop(Mal-Activity_3,
"broadcastADSBPacket"^^string),
isActorMalicious(?h1, true)
```

Listing 2

The main idea here is that if the ATC Server receives a packet from a helicopter, then it computes its corresponding hash based on the packet’s payload and compares it to the hash received in the packet. We assume that the ADS-B packet contains a hash in its payload. If these values match, then the ATC Server proceeds with broadcasting the packet. Otherwise, it marks the helicopter that sent the forged packet as malicious and the “mitigates” object property is proven to be valid. In order to prove this theorem, the Pellet reasoner checks the data and object properties of the individuals concerned by this rule and tries to infer the head of the rule. In this case, if it succeeds in doing so, the object relations in the head appear as highlighted and we can get the explanation of the proof.

V. Classification of Attacks Using Metrics

In this section, we describe the taxonomy we have developed for the message injection ADS-B attacks. It is composed by three classes of attacks, classified based on the difficulty of implementation and the location of the radio device that was used by the attacker. These classes are as follows:

- **Medium-level attacks:** in this type of attacks, the attacker generates the malicious ADS-B messages to be injected in a typically random way and he does not move the equipment used to launch the attack. For instance, the attacker can send a massive amount of ADS-B messages whose locations are within the reach of the ATC Sever with fake aircraft identifier in order to obstruct the view of the radar display and thus prevent the air traffic controller from performing his duties.

- **Advanced-level attacks:** where an attacker uses sophisticated flight simulator programs along with the radio device in order to send a more realistic flight path that cannot be detected as fake easily. For example, one popular program that can be used to achieve this is FlightGear [18]. In this case, the location of the equipment used to perform the attack is fixed.

- **Expert-level attacks:** similar to the advanced-level attacks, except for the fact that the equipment used to launch the attack is located in an aircraft. This kind of attack is harder to test, since it requires sophisticated equipment and procedures.

Classifying attacks detected using the techniques described in the previous section requires collecting parameters, needed for deciding if an attack belongs to a particular attack class modelled in an ontology rule. We leveraged the work in [12], a well-known standard that provided us with a reference for checking under which category our metrics fell into. We defined three security metrics, which are described as follows:

- **Sender Location Difference**: the absolute value of the difference between the triangulated sender's location at two consecutive times t_i and t_j. We assumed having appropriated means of triangulation, which is needed for determining the location of the sender based on the received ADS-B packet. This metric is broken down into three sub-metrics which correspond to the differences between longitudes, latitudes, and altitudes.

- **Velocity**: which is the speed of the aircraft at a time *t*.

- **Estimated-to-Real Difference**: which is the absolute value of the difference between the estimated location of an aircraft and the location retrieved from the ADS-B packet at time *t*. We assumed the capability of estimating aircraft locations at all times. This metric is also composed by three sub-metrics, corresponding to difference of longitudes, latitudes and altitudes.

After careful consideration, we came to a conclusion that these three defined metrics fell under the Cyber Intelligence Threat Analysis category. After all, these metrics collect practical data about the attacks, and allow the security analyst to classify cyber attacks based on patterns of wrong behavior. In our methodology, these metrics are used by the Pellet reasoner to automatically classify the type of attack. The relationships between the described classes of attacks and the security metrics are described as follows:

- **Medium-level attacks**: an attack belongs to this class if the sender location difference and the velocity are equal to zero. An attacker, whose physical location does not change, is of course very likely to have such characteristics. Further, the estimated-to-real difference has to be greater than a predefined threshold for the longitude, latitude and altitude. Consequently, if the location retrieved from the ADS-B packet is not within the aircraft envelope, then such packet most likely comes from an attacker.
- **Advanced-level attacks**: An attack belongs to this class if the sender location difference and the velocity are equal to zero. The estimated-to-real difference would be within the predefined thresholds.
- **Expert-level attacks**: An attack belongs to this class if the velocity is comparable to the one of a real aircraft. Besides, the sender location difference cannot be equal to zero, and the estimated-to-real difference has to be within the predefined thresholds.

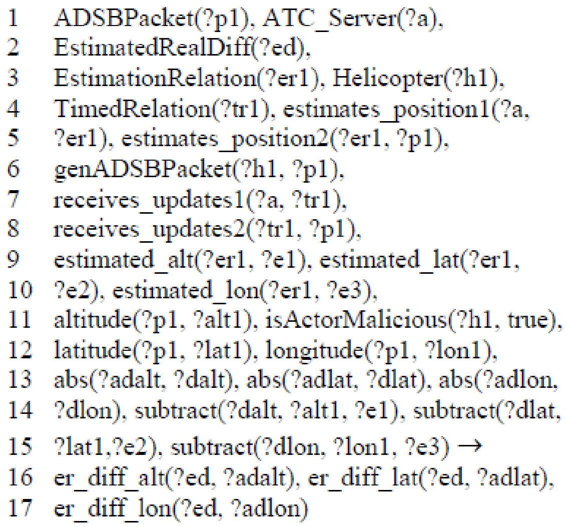
We now focus on how the proofs of the theorems are generated using ontological rules, similarly to the previous section of this paper. Due to space limitations, we restrict the explanation to the ontological rules used for computing the estimated-to-real difference metric, and for classifying an attack as belonging to medium-level attacks class respectively in Listings 3 and 4.

In Listing 3, we collect GPS properties of the malicious actor, after verifying that the packet he sent had reached the ATC Server. Then, we compute the properties of the estimated-to-real metric relatively to longitude, latitude and altitude.

In Listing 4, we collect the data provided by all the metrics and define the conditions for deciding whether an observed attack belongs to the medium-level attacks class.

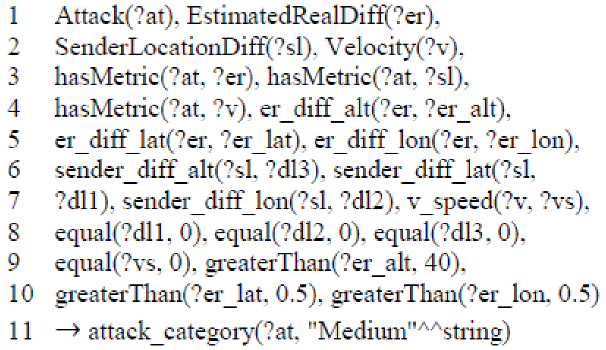
We had to make several assumptions regarding the metrics. Firstly, we assumed that there is a mechanism to triangulate the true location of the sender of the packet, which would facilitate computing its location difference metric. Secondly, we assumed that it is possible to estimate the location of an aircraft at all times, which is required for computing the estimated-to-real difference metric.

For each metric used in this evaluation we have written a corresponding rule that the Pellet reasoner can use against the defined individuals to determine its value. The first rule is related to the sender location difference metric. Basically, it gets the triangulated sender locations at two consecutive time steps and calculates the absolute value of the difference of altitude, latitude and longitude. The second rule addresses the velocity metric, and extracts its value at a certain time by using the *ADSBPacket* and *TimedRelation* entities. The third rule,



```
ADSBPacket(?p1), ATC_Server(?a),
EstimatedRealDiff(?ed),
EstimationRelation(?er1), Helicopter(?h1),
TimedRelation(?tr1), estimates_position1(?a,
?er1), estimates_position2(?er1, ?p1),
genADSBPacket(?h1, ?p1),
receives_updates1(?a, ?tr1),
receives_updates2(?tr1, ?p1),
estimated_alt(?er1, ?e1), estimated_lat(?er1,
?e2), estimated_lon(?er1, ?e3),
altitude(?p1, ?alt1), isActorMalicious(?h1, true),
latitude(?p1, ?lat1), longitude(?p1, ?lon1),
abs(?adalt, ?dalt), abs(?adlat, ?dlat), abs(?adlon,
?dlon), subtract(?dalt, ?alt1, ?e1), subtract(?dlat,
?lat1,?e2), subtract(?dlon, ?lon1, ?e3) →
er_diff_alt(?ed, ?adalt), er_diff_lat(?ed, ?adlat),
er_diff_lon(?ed, ?adlon)
```

Listing 3



```
Attack(?at), EstimatedRealDiff(?er),
SenderLocationDiff(?sl), Velocity(?v),
hasMetric(?at, ?er), hasMetric(?at, ?sl),
hasMetric(?at, ?v), er_diff_alt(?er, ?er_alt),
er_diff_lat(?er, ?er_lat), er_diff_lon(?er, ?er_lon),
sender_diff_alt(?sl, ?dl3), sender_diff_lat(?sl,
?dl1), sender_diff_lon(?sl, ?dl2), v_speed(?v, ?vs),
equal(?dl1, 0), equal(?dl2, 0), equal(?dl3, 0),
equal(?vs, 0), greaterThan(?er_alt, 40),
greaterThan(?er_lat, 0.5), greaterThan(?er_lon, 0.5)
→ attack_category(?at, "Medium"^^string)
```

Listing 4

which deals with the estimated-to-real difference, gets the coordinates of both the estimated position and the real position before calculating the absolute value of the difference in terms of latitude, longitude and altitude.

The knowledge derived from these rules can provide reasonable insights into attack classification. We developed different rules to classify an attack according to its category in the taxonomy. For example, an attack that belongs to the medium-level category would have a null velocity, a null sender location difference, and its estimated-to-real metric would exceed the defined threshold. Conversely, the advanced-level category would have its attacks with a null velocity and a null sender location difference, but its estimated-to real metric would not exceed the defined threshold. This is expected, given the use of flight simulator versus generating random values in the medium-level category. Finally, an attack would be in the expert-level category if the velocity is comparable to a real aircraft, while its sender location difference would be greater than zero and its estimated-to-real metric would not exceed the pre-defined threshold.

VI. Related Work

In [7], Massacci et al. proposed an ontology for security requirements by extending existing ontologies with situational and context awareness. The authors contextualize their ideas by an ADS-B case study. This work is similar to ours but the main difference is that they focused on GPS spoofing attacks, while we address message-injection attacks that are more difficult to realize, as stated by [3].

Oltramari et al. [19] described an approach to support cyber operations by enhancing the situational awareness via a combination of cognitive modelling and ontology engineering. They plan to evaluate their approach by applying it to design a cyber defense application. However, their work is not specific to SDR applications, but to cyber operations in general.

In [8], Obrst et al. presented a methodology for building cyber security ontologies based on a malware ontology. This methodology outlines the steps that are required for building a cyber security ontology, and provide general guidelines for enhancing the cyber security domain with semantic models. The main difference between this work and ours is their focus on security from a general standpoint, starting from a wide characterization of malware. In our paper, we tackle the problem of security within the SDR domain by leveraging knowledge from semantic models and ontologies.

In [9], Ekelhart et al. introduced a framework for building security ontologies that assists in providing risk analysis. The authors used an incremental approach where they start with a generic security taxonomy formalized in an ontology and they enhance it by integrating risk factors, constraints, threats and countermeasures. This work concentrates on risk management involving IT-security tasks in a company, while our goal is to create a methodology to secure ADS-B networks.

In [20], Magklaras and Furnell proposed an approach to address internal IT misuse via a classification of misusers and their motives, as well as the implications of the misuse on the system. In our paper, we adopted a more flexible representation of misuses, which relies on misuse case and mal-activity diagrams. Moreover, their work describes security in general while ours focuses on security in ADS-B networks. The authors provided a mechanism of determining the threat level that is similar to our work, in which we classify the attacks according to the taxonomy. The main difference is that we employ theorem proving with a semantic-web inspired rule system, while their work is based on an analyzer module built as part of their proposed framework.

In [21], McCallie et al. assessed ADS-B security by detecting and classifying attacks that may target ADS-B applications. They provide some general recommendations on how to thwart these attacks. In contrast, we provide a methodology to be applied when analyzing the security of SDR applications.

Similarly, Costin and Francillon [4] demonstrated the lack of security of ADS-B by implementing attacks in a low-cost simulated environment. They did not focus on attack mitigation. In contrast, our methodology assists the systems engineer in formulating security requirements by precisely defining and verifying these for SDR applications, while using automated design verification for attacks and their mitigations.

In [22], Whittle et al. proposed a technique for modeling possible attacks and mitigating them. They employ misuse cases to model undesirable system behavior. The approach models misuse cases as aspects, inserts these in the core system features before integrating mitigation techniques. Then, they use the attacks as test cases to evaluate the design robustness. Although our objective is similar to theirs, but we base our methodology on ontologies to support the system design from the ground up with security as an integral design aspect. In contrast, they use prior work on state machines.

In [23], Sindre introduced the concept of mal-activity diagrams as an enhanced form of activity diagrams where each actor of the system, normal or malicious, occupies a swimlane and starts normal or malicious activity nodes. Our approach uses the concept of a mal-activity diagram and integrates it in the design process with the support of ontologies.

In [24], El-Attar presented a tool to convert a textual description of the system to a model taking into consideration the security aspects in term of misuse case and mal-activity diagram. This is achieved with support from two tools. One transforms the textual description to a context-free grammar, which is used to build the first meta-model. The other creates the meta-model that captures the mal-activity diagrams. This work appears similar to ours, but El-Attar's main goal is to create meta-models from textual description. In contrast, we formally capture the diagrams using ontological rules and verify that the stated relationships between them exist using a theorem prover.

VII. Conclusion

ADS-B has emerged as a promising technology for optimizing the use of the air space while lowering costs and increasing the security of air traffic operations. Hindering this progress, many security vulnerabilities in the protocol have been discovered, generating a pressing need for a holistic, systems-oriented approach to properly address the problem. Within this context, in this paper we present a methodology that relies on time-tested, traditional requirements engineering while leveraging advanced semantic technology concepts to automate the process of requirement verification. We have tested the methodology in an ADS-B scenario, and were able to support the system design by translating security requirements into formally verifiable claims. Finally, we used logical reasoning to ascertain the validity of the mitigating solutions and classify the attacks using security metrics.

We plan to further evaluate the methodology in complex simulation environments that will provide a better understanding of its broader impact in designing resilient SDR applications. Future work on the methodology also involves standardizing its procedures, so they would be applicable to the field of SDR applications in a consistent fashion. In this paper we have focused on the initial phases of the system engineering life-cycle, but the methodology can be easily expanded to formalize and automate other parts of the systems engineering life cycle. Examples of the latter include supporting trade-off analysis of adding security features against their associated cost, validation and verification of the actual system based on stakeholder requirements (e.g. FAA specs for different types of systems), and others that would benefit from the formalization of the design process with a focus on its security requirements.

REFERENCES

[1] J. Mitola, "The software radio architecture," *IEEE Communications Magazine*, vol. 33, no. 5, pp. 26–38, May 1995.

[2] "NASA - NextGen," 24-Oct-2014. Available at: http://www.hq.nasa.gov/office/aero/asp/airspace/index.htm.

[3] M. Strohmeier, V. Lenders, and I. Martinovic, "On the Security of the Automatic Dependent Surveillance-Broadcast Protocol," *arXiv:1307.3664 [cs]*, Jul. 2013.

[4] A. Costin and A. Francillon, "Ghost in the Air(Traffic): On insecurity of ADS-B protocol and practical attacks on ADS-B devices," 2012.

[5] M. Schäfer, V. Lenders, and I. Martinovic, "Experimental Analysis of Attacks on Next Generation Air Traffic Communication," in *Applied Cryptography and Network Security*, Springer Berlin Heidelberg, 2013, pp. 253–271.

[6] G. Sindre and A. L. Opdahl, "Eliciting security requirements with misuse cases," *Requirements Eng*, vol. 10, no. 1, pp. 34–44, Jan. 2005.

[7] F. Massacci, J. Mylopoulos, F. Paci, T. T. Tun, and Y. Yu, "An Extended Ontology for Security Requirements," in *Advanced Information Systems Engineering Workshops*, C. Salinesi and O. Pastor, Eds. Springer Berlin Heidelberg, 2011, pp. 622–636.

[8] L. Obrst, P. Chase, and R. Markeloff, *Developing an Ontology of the Cyber Security Domain*. .

[9] A. Ekelhart, S. Fenz, M. Klemen, and E. Weippl, "Security Ontologies: Improving Quantitative Risk Analysis," presented at the 40th Annual Hawaii International Conference on System Sciences, 2007. HICSS 2007, 2007, p. 156a–156a.

[10] H. Knublauch, R. W. Fergerson, N. F. Noy, and M. A. Musen, "The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications," in *The Semantic Web – ISWC 2004*, S. A. McIlraith, D. Plexousakis, and F. van Harmelen, Eds. Springer Berlin Heidelberg, 2004, pp. 229–243.

[11] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical OWL-DL reasoner," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, no. 2, pp. 51–53, Jun. 2007.

[12] "MITRE's Making Security Measurable," *MITRE's Making Security Measurable*. Available: http://makingsecuritymeasurable.mitre.org/.

[13] "ADS-B Frequently Asked Questions (FAQs)," 07-Apr-2014. Available at: http://www.faa.gov/nextgen/implementation/programs/adsb/faq/#3.

[14] D. Magazu III, "Exploiting the Automatic Dependent Surveillance-Broadcast System via False Target Injection," 2012.

[15] *GNU Radio*. Available at www.gnuradio.org.

[16] N. Foster, "Gr-air-modes," Available: https://github.com/bistromath/gr-air-modes.

[17] I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML," May 2004.

[18] "FlightGear." Available: www.flightGear.com.

[19] A. Oltramari, C. Lebiere, W. Zhu, L. Vizenor, and R. Dipert, "owards a Cognitive System for Decision Support in Cyber Operations," presented at the International Conference on Semantic Technologies for Intelligence, Defense, and Security (STIDS), 2013.

[20] G. B. Magklaras and S. M. Furnell, "Insider Threat Prediction Tool: Evaluating the probability of IT misuse," *Computers & Security*, vol. 21, no. 1, pp. 62–73, Jan. 2001.

[21] 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, no. 2, pp. 78–87, Aug. 2011.

[22] J. Whittle, D. Wijesekera, and M. Hartong, "Executable misuse cases for modeling security concerns," presented at the ACM/IEEE 30th International Conference on Software Engineering, 2008. ICSE '08, 2008, pp. 121–130.

[23] G. Sindre, "Mal-Activity Diagrams for Capturing Attacks on Business Processes," in *Requirements Engineering: Foundation for Software Quality*, P. Sawyer, B. Paech, and P. Heymans, Eds. Springer Berlin Heidelberg, 2007, pp. 355–366.

[24] M. El-Attar, "From misuse cases to mal-activity diagrams: bridging the gap between functional security analysis and design," *Software Systems Modelling*, vol. 13, no. 1, pp. 173–190, Feb. 2014.

An Ontology for Insider Threat Indicators

Development and Applications

Daniel L. Costa, Matthew L. Collins, Samuel J. Perl, Michael J. Albrethsen, George J. Silowash, Derrick L. Spooner
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA, USA
insider-threat-feedback@cert.org

***Abstract*—We describe our ongoing development of an insider threat indicator ontology. Our ontology is intended to serve as a standardized expression method for potential indicators of malicious insider activity, as well as a formalization of much of our team's research on insider threat detection, prevention, and mitigation. This ontology bridges the gap between natural language descriptions of malicious insiders, malicious insider activity, and machine-generated data that analysts and investigators use to detect behavioral and technical observables of insider activity. The ontology provides a mechanism for sharing and testing indicators of insider threat across multiple participants without compromising organization-sensitive data, thereby enhancing the data fusion and information sharing capabilities of the insider threat detection domain.**

Keywords—ontology; insider threat; data fusion; information sharing

I. Background

The study of insider threat presents some of the most complex challenges in information security. Even defining the insider threat has proven difficult, with interpretations and scope varying depending on the problem space. The CERT® Division of Carnegie Mellon University's Software Engineering Institute defines a malicious insider as a current or former employee, contractor, or other business partner who has or had authorized access to an organization's network, system, or data and intentionally exceeded or misused that access in a manner that negatively affected the confidentiality, integrity, or availability of the organization's information or information systems [1]. Organizations have begun to acknowledge the importance of detecting and preventing insider threats, but there is a surprising lack of standards within the insider threat domain to assist in the development, description, testing, and sharing of these techniques. For many organizations, establishing an insider threat program and beginning to look for potentially malicious insider activity is a new business activity. In particular, Executive Order 13587 and the National Insider Threat Policy describe minimum standards for establishing an insider threat program and monitoring employee use of classified networks for malicious activity [2-4].

II. Purpose

A. Goals

The primary goal of this effort is to support the creation, sharing, and analysis of indicators of insider threat. Because insider data is sensitive, insider threat teams often work only with data from inside their own organizations. These records frequently include documented employee behaviors, intellectual property, employee activity on networks, and information on organizational proprietary networks and information technology (IT) architecture. Organizations and teams are hesitant to release this information due to the risk of breaching employee privacy, releasing sensitive organizational information, or unnecessarily losing a competitive advantage. A shared ontology will allow teams to share indicators of insider threat without disclosing their own sensitive data. Our desired outcome is to facilitate information sharing on effective indicators of malicious insider activity across organizations, with an emphasis on extensibility, semi-automation, and the ability for community members to benefit from investigations and analysis performed by others.

B. The Case for an Ontology

All entity and relationship data models, including semantic data models, have their limitations [5]. Models are extremely formal by design and can encounter problems when representing the variety of actions involved in a real-world insider threat case. In addition, the data on cases of insider threat is often gathered from legal judgments and outcomes whose documentation is highly variable. As a result, insider threat domain experts tend to rely on natural language to document their cases and findings. Though natural language is more expressive than a model, we believe the insider threat domain will benefit from the development of an ontology. Our interest in building an ontology, developed from our observations of the field today, is driven by the following factors:

- We expect rapid growth in the data being collected and shared by organizations, specifically about insider threats. Some organizations have already stated that overcoming this challenge is one of their top priorities [6].
- The insider threat research community lacks a defined, formal model that is machine readable, human understandable, and transferrable with limited sharing

barriers. We felt that starting a model of this kind, based on the real-world case data we have already collected, could accelerate this process within the community, as has been done in other fields [7, 8].

- We are willing to accept some loss of descriptive power for individual cases, provided we can analyze large populations of cases using computation. We expect insider threat teams (both in research and in operations) to be asked to detect insider threat activity by analyzing a growing quantity of data from new sources in an increasingly limited amount of time.

III. Approach

A. Domain Identification

At first glance, defining the domain of our ontology appeared to be a trivial matter: representation of potential indicators of malicious insider activity. In practice, indicators of malicious insider activity involve complex interconnections of parts of several other domains:

- Human behavior: understanding insider threats involves understanding the people behind the malicious activity—the reasons why they attacked, their psychological characteristics, their emotions, and their intent.
- Social interactions and interpersonal relationships: modeling the relationships between insiders and their employers, colleagues, friends, and family is a crucial part of identifying stressors that are often associated with malicious insider activity.
- Organizations and organizational environments: the culture and policies of organizations factor heavily into the interpretation of malicious behavior within an organization.
- Information technology security: information and information systems can be both the targets of and tools used to perpetrate malicious insider activity. IT security also contains other concepts of interest in describing the insider threat domain, namely, confidentiality, integrity, and availability.

B. Domain Scoping

With a representative list of sub-domains for insider threat enumerated, our next challenge was determining the scope at which our ontology must provide support for each subdomain. We chose to develop the following competency questions for our ontology to assist us in our scoping efforts [9, 10].

- What concepts and relationships comprise the technical and behavioral observables of potential indicators of malicious insider activity?
- What potential indicators of malicious insider threat activity are insider threat teams using for detection?
- To facilitate information sharing, at what level of detail should organizations describe their indicators of malicious insider activity without revealing organization-sensitive information?

C. Construction Method

Since 2001, the CERT® Insider Threat Center has collected over 800 cases in which insiders used IT to disrupt an organization's critical IT services, commit fraud against an organization, steal intellectual property, or conduct national security espionage, sabotaging systems and data, as well as other cases of insiders using IT in a way that should have been a concern to an organization. This data provides the foundation for all of our insider threat research, our insider threat lab, insider threat assessments, workshops, exercises, and the models developed to describe how the crimes evolve over time. Our case collection involves gathering and analyzing data from public (e.g., media reports, court documents, and other publications) and nonpublic (e.g., law enforcement investigations, internal investigations from other organizations, interviews with victim organizations, and interviews with convicted insiders) sources. This data collection, summarized in Figure 1, primarily focuses on gathering information about three entities: the organizations involved, the perpetrator of the malicious activity, and the details of the incident. Each case in our insider incident repository contains a natural language description of the technical and behavioral observables of the incident. We used these descriptions as the primary data source for our ontology.

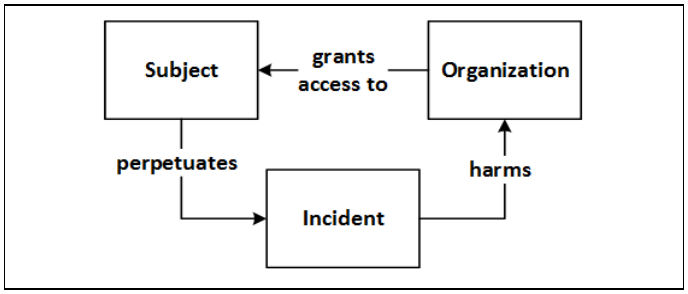


Fig. 1. CERT model for insider incidents

1) Data-Driven Ontology Bootstrapping

To ensure full coverage of the information contained in our insider incident repository, we adopted an approach that utilizes concept maps as a first step in the development of an ontology [11]. Manually developing concept maps for over 800 individual insider threat cases required an infeasible level of effort, so we developed a semi-automated concept map extraction method adapted from several existing approaches [12, 13]. This method used part-of-speech and part-of-sentence tagging to extract [concept, concept, relationship] triples from the natural language description of each insider incident. We utilized additional text and natural language processing techniques to eliminate stop-words, group similar triples, and sort the triple collection by frequency of occurrence. We then used this collection of triples as the basis for our class hierarchy, using our competency questions to set scope and optimize the arrangement of specific classes.

2) Additional Data Sources

We supplemented the candidate classes and object properties derived from our insider incident repository with concepts and relations from the cyber threat and digital forensics domains. We reviewed the Structured Threat

Information Exchange (STIX) and Cyber Observable Expression (CybOX) languages [14, 15], as well the SANS Institute's digital forensics artifact catalog [16], to fill gaps in our concepts for cyber threats, cyber observables, and their associated forensic artifacts.

IV. Implementation

A. *Design Decisions*

We adapted components from several existing ontologies for our work. To assist in the modeling of actors and their actions, we adapted several top-level ontology components from material available on schema.org [17]. We leveraged existing ontologies for filling gaps in our coverage of cyber assets, including concepts from the network services, IT systems, IT security, and mobile device domains [18-21]. To validate our design, we used the catalog of common ontology development pitfalls from work titled "Validating ontologies with oops!" [22]. We provided support for modeling the temporality of actions and events relative to one another through use of the sequence design pattern [23]. We have chosen to implement our ontology using the Web Ontology Language (OWL), due to its maturity, wide use, and extensibility [24].

B. *Overview of Top-Level Classes*

The top-level of our ontology, summarized in Figure 2, is composed of five classes: Actor, Action, Asset, Event, and Information. The Actor class contains subclasses for representing people, organizations, and organizational components such as departments. The Action class contains the subclasses that define the things that actors can perform. The Asset class provides subclasses that define the objects of actions. The Information class provides subclasses that provide support for modeling the information contained within some assets (examples include personally identifiable information, trade secrets, and classified information). The Event class provides support for multiple types of events of interest. Events are generally associated with one or more Actions. The creation of an individual event typically requires making some inference, as opposed to an individual Action, which can be created through direct observation. For example, moving a file is modeled in our ontology as an Action. A data exfiltration event, when associated with a file move action via the *hasAction* object property, expresses the fact that the associated action was unauthorized. Additionally, an object property hierarchy is provided to express various types of relationship roles, job roles, and event roles.

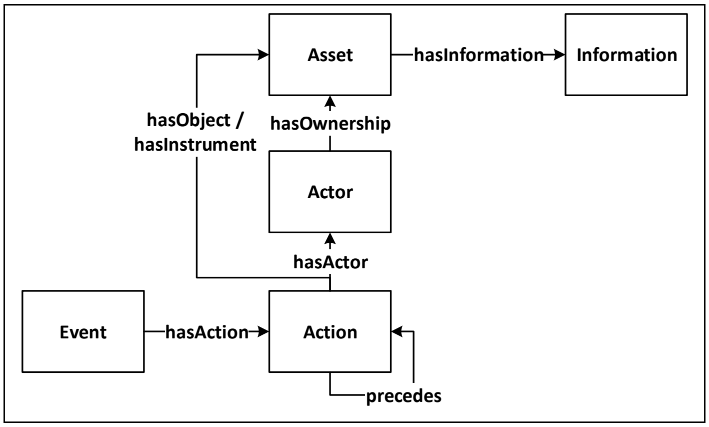


Fig. 2. Top-level ontology classes and object properties

C. *Example Uses*

To demonstrate use of the ontology to describe indicators of malicious insider activity, we present two examples of translating natural language descriptions of indicators of malicious insider activity from our insider threat incident repository into ontology individuals. The translation process is relatively straightforward; the concepts from each description are manually identified, individuals are created for each concept as instances of the appropriate ontology class, and individual object properties are added to relate the class instances to one another. Figure 3 and Figure 4, respectively, depict the ontology translation for the following insider threat indicator descriptions:

- The insider transferred proprietary engineering plans from the victim organization's computer systems to his new employer.
- The insider accessed a web server with an administrator account and deleted approximately 1,000 files.

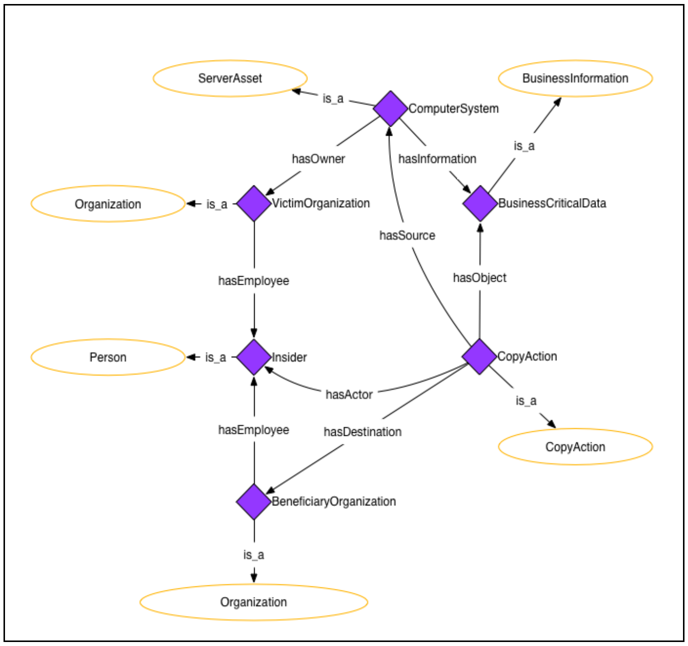


Fig. 3. Data exfiltration example from insider incident repository translated into ontology individuals

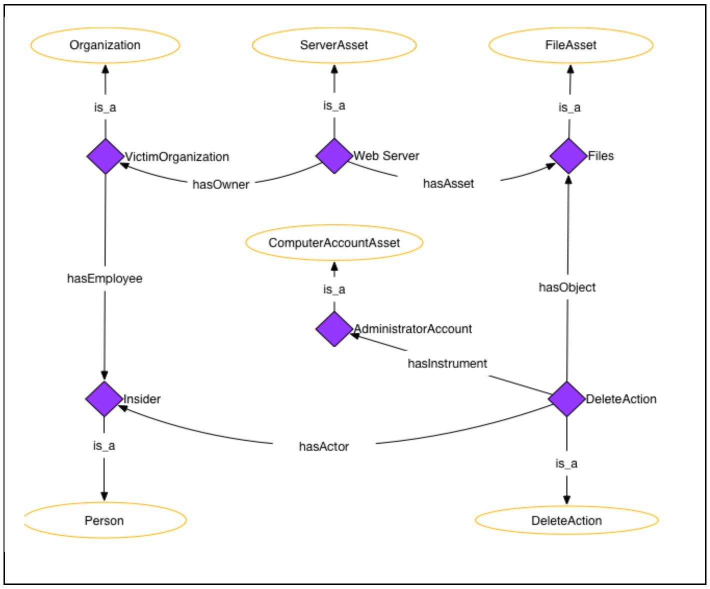


Fig. 4. Information technology sabotage example from insider incident translated into ontology individuals

V. Applications

A. *Insider Threat Indicator Information Sharing*

Our ontology provides two powerful concepts in the description of potential indicators of malicious insider activity: abstraction and extensibility. By abstraction, we mean that indicators can now be described at a level of detail that omits organization-sensitive information while still maintaining enough descriptive information to express the idea that given observable actions or conditions are potential indicators of malicious insider activity. By extensibility, we mean that we have provided the conceptual components that organizations can use to describe their existing indicators and develop new indicators. Potential indicators of malicious insider activity often include qualifiers such as “excessive,” “anomalous,” “unauthorized,” and “suspicious” to distinguish conditions that are potentially indicative of malicious insider activity from “normal” behavior and activity. Definitions and interpretations of these types of conceptual qualifiers vary greatly from organization to organization, and often vary within organizations based on variables such as job type, location, and time. To accommodate these variations, we introduce the idea of “policy packs” in our ontology: modular collections of ontology axioms that represent organization-agnostic concepts, definitions, and interpretations of indicator patterns. Our ontology specifically provides support for this via the Event class hierarchy. Organizations using our ontology can develop their own defined classes, or modify existing ones, to specify the necessary and sufficient restrictions for class membership.

B. *Automated Indicator Instance Extraction Framework*

Insider threats can be detected by observing instances of indicators of malicious insider activity within an organization. Operationally, this involves the collection and analysis of large amounts of data on every employee in an organization. Without some level of automation, this detection practice becomes infeasible to perform effectively and efficiently. Using our ontology, we have designed a semi-automated approach for the detection of potential indicators of malicious insider activity that fuses data from multiple types of sources. The ontology provides an analysis hub that combines information from an organization’s enterprise network activity and human resources data to provide a data-rich environment for the development and detection of robust, effective indicators of malicious insider activity.

1) Operational Data to Ontology Individuals

We use the term “operational data” to encapsulate the data and data sources that capture the user-based activity that occurs on an organization’s information systems and networks. The technical observables associated with some potential indicators of malicious insider activity are found in operational data and during the analysis of trends in operational data. Some examples of operational data include:

- Host-based user activity logs
- Critical application audit logs
- Network activity logs
- Communication server logs
- System event logs

Since operational data is usually found in structured or semi-structured log files, we attempted to prove the concept of automatically translating the information contained in operational data sources into ontology individuals. Instead of direct translation into ontology individuals from operational data sources, we chose to translate the operational data into CybOX cyber observable files, and automatically create ontology individuals based on the contents of the CybOX files. This approach allowed us to focus on identifying the fields from CybOX that were applicable to our ontology classes, and provide a translation mechanism for only those applicable fields. Without the CybOX translation layer, we would have had to develop ontology translation mechanisms for each type of operational data source we wish to support, which would require an infeasible level of effort, support, and maintenance. Additionally, CybOX provides an API for their XML file format, which facilitates the automated translation of any input data source into the CybOX format. (CybOX currently supports over 60 input data sources.)

In our proof of concept, we were successful in automatically translating Windows system event logs into the CybOX format, and, using simple scripts, automatically generating the OWL XML code to create individuals for a small subset of our ontology classes. In a robust implementation, the automated ontology individual creation would provide configurable settings that would allow organizations to control the creation of ontology individuals for classes whose specific definitions may vary from organization to organization. For example, if the ontology contained a class representing after-hours logins, the automated individual creation mechanism should provide a way to specify a time range that is considered after-hours.

2) Human Resources Data to Ontology Individuals

We use the term “human resources data” to encapsulate data and data sources that provide contextual and behavioral information about employees. These records are typically stored in an unstructured format, and are locked within Human Resources departments to protect the privacy rights of employees. Examples of human resources data include:

- Organization charts
- Employee performance reviews
- Employee personnel files, including job title, supervisor, role, and responsibilities
- Employee behavior records, such as formal reprimands and policy violations
- Information from anonymous insider reporting channels
- Results of background checks

Human resources data provides a rich source of contextual, behavioral, and psychosocial information regarding employees. Human resources data is typically more fragmented and less structured than operational data, so the automated translation of this data into ontology individuals may be a challenge for some organizations. Enterprise solutions for human resource information management exist, and where they are used, a structured representation of human resources data could be used to develop an automated ontology translation process. In our proof of concept for the automated indicator instance extraction framework, we did not attempt to automatically create ontology individuals from human resources data, but in future work, we will apply a similar approach to we used for operational data.

3) Semantic Reasoner

If operational data and human resources data are both described using the ontology, and if indicator policy packs are in place, an organization can use a semantic reasoner to make inferences and automatically classify ontology individuals as instances of specific defined classes. Ontology individuals that meet the formal definitions of potential indicators of malicious insider activity can then be said to have “satisfied” some indicator. A collection of ontology individuals that satisfy threat indicators becomes a useful data set for insider threat analysts. With a robust set of indicators implemented as defined classes, analysts have the ability to see descriptions of potential indicators of malicious insider activity across previously disparate data sets and at larger scale. Satisfied indicators can be reviewed by analysts to identify false positives, refine indicators, develop new indicators to add back into the ontology via policy packs, or create threat reports that summarize the potential malicious insider activity found in the data.

4) Putting it All Together

The full framework—beginning with the development and maintenance of the ontology through the release of organizational threat reports based on the detected instances of potential indicators of malicious insider activity—is presented in Figure 5. This framework is meant to support detection of potential indicators of malicious insider activity that is then triaged. An effective implementation of the framework depends on the indicators it contains, and not all satisfied indicators necessarily warrant an investigation.

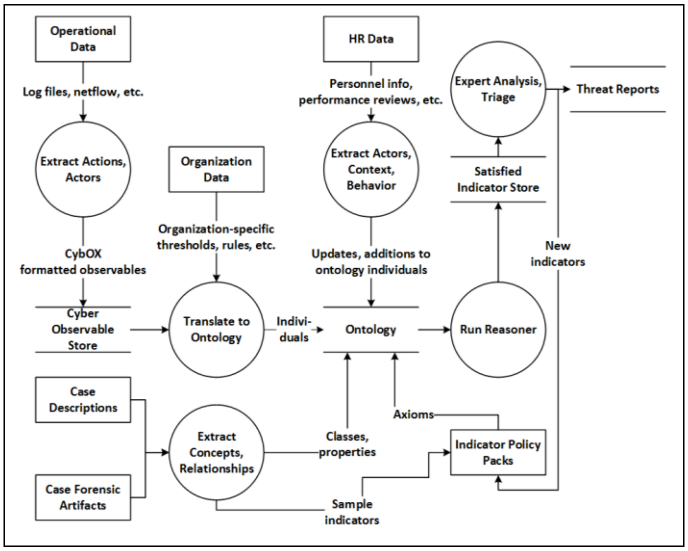


Fig. 5. Data flow diagram for automated indicator instance extraction framework

The evaluation of specific instances of indicators requires expert analysis and investigation to remove false positives, assess severity of the satisfied indicator, and perform set and temporal analysis on the satisfied indicators. The framework can support a workflow-based analysis and incident escalation process. Specific implementations of the framework are expected to grow and change as the organization, its insider threat program, and the larger insider threat community and domain all do the same. The activities associated with the operations and maintenance of this framework include

- Identifying new candidate indicators during the analysis of satisfied indicators
- Adding new indicators to the ontology as updates or additions to indicator policy packs
- Re-running the semantic reasoner as new ontology individuals are created and new indicators are added
- Adding automated ingest support for new operational data sources
- Extending the human resources data ingest process to include new data sources
- Updating the configuration for the automated ontology individual extractor as organizational policies change and new insights are gained

In addition to the activities mentioned above, the ontology itself will grow and change over time. The drivers for ontology changes will be the addition of new concepts and relationships based on analysis of new cases involving malicious insider activity, as well as feedback from the organizations that are using the ontology. Finally, indicator policy packs can be safely shared with other organizations as a means of identifying effective industry specific and domain-wide detection strategies and patterns.

VI. Conclusion

With the initial development of our ontology, we have created a bridge between natural language descriptions of potential indicators of malicious insider activity in case data and the operational data that contains the technical and behavioral observables associated with malicious insider activity. We have provided a mechanism that allows sensitive information to be abstracted away while maintaining enough descriptive ability to effectively communicate actions and behaviors of interest across organizations. By introducing the application of our ontology as an analysis hub that combines operational and human resources data, we have laid the foundation for more effective fusion of these traditionally disparate data sources.

VII. Future Work

As we continue the development of our ontology, we will perform the following activities in future work:

- Provide enhanced support for behavioral components of potential indicators of malicious insider activity
- Collaborate with other organizations to improve the expression of insider threat indicators using the ontology
- Add support for additional indicator policy packs
- Mature the proof of concept automated indicator instance extractor and provide customization options for additional data sources and organization configurations
- Assess the feasibility of automating the creation of ontology individuals based on human resources data
- Evaluate formal ontology validation methods and apply them to our ontology

Acknowledgement

The authors gratefully acknowledge support for this work from the Defense Advanced Research Projects Agency (DARPA) and the Federal Bureau of Investigation. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Approved for Public Release, Distribution Unlimited.

This material is based upon work funded and supported by Federal Bureau of Investigation under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense.

References

[1] D. M. Cappelli, A. P. Moore, and R. F. Trzeciak, *The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes (Theft, Sabotage, Fraud)*: Pearson Education, 2012.

[2] U.S. GOVERNMENT, "Executive Order 13587-Structural Reforms To Improve the Security of Classified Networks and the Responsible Sharing and Safeguarding of Classified Information," 2011.

[3] B. Obama, "National Insider Threat Policy and Minimum Standards for Executive Branch Insider Threat Programs," T. W. House, Ed., ed: Office of the Press Secretary, 2012, p. 1.

[4] F. o. A. Scientists, "National Insider Threat Policy and Minimum Standards for Executive Branch Insider Threat Programs (Minimum Standards)," T. W. Hourse, Ed., ed. www.fas.org: Federation of American Scientists, 2012.

[5] M. West, *Developing high quality data models*: Elsevier, 2011.

[6] F. Intelligence and National Security Alliance (INSA) in partnership with DHS, and ODNI. (2014). Insider Threat Resource Directory. Available: http://www.insaonline.org/insiderthreat

[7] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, et al., "Gene Ontology: tool for the unification of biology," *Nature genetics*, vol. 25, pp. 25-29, 2000.

[8] S. Schulze-Kremer, "Adding semantics to genome databases: towards an ontology for molecular biology," in Ismb, 1997, p. 5.

[9] M. Grüninger and M. S. Fox, "The role of competency questions in enterprise engineering," in Benchmarking—Theory and Practice, ed: Springer, 1995, pp. 22-31.

[10] A. Gangemi, "Ontology design patterns for semantic web content," in The Semantic Web–ISWC 2005, ed: Springer, 2005, pp. 262-276.

[11] R. R. Starr and J. M. P. de Oliveira, "Conceptual maps as the first step in an ontology construction method," in Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2010 14th IEEE International, 2010, pp. 199-206.

[12] K. Žubrinic, "Automatic creation of a concept map."

[13] J. J. Villalon and R. A. Calvo, "Concept Map Mining: A definition and a framework for its evaluation," in Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on, 2008, pp. 357-360.

[14] S. Barnum, "Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIX™)," *MITRE Corporation, July*, 2012.

[15] The MITRE Corporation. (2014). *Cyber Observable eXpression*. Available: http://cybox.mitre.org/language/version2.1/

[16] R. Lee, "SANS Digital Forensics and Incident Response Poster Released," in Blog: SANS Digital Forensics and Incident Response Blog vol. 2014, S. D. Faculty, Ed., ed. SANS: SANS, 2012.

[17] *schema.org*. Available: http://schema.org

[18] J.-b. Gao, B.-w. Zhang, X.-h. Chen, and Z. Luo, "Ontology-based model of network and computer attacks for security assessment," *Journal of Shanghai Jiaotong University (Science)*, vol. 18, pp. 554-562, 2013/10/01 2013.

[19] S. Fenz and A. Ekelhart, "Formalizing information security knowledge," in Proceedings of the 4th international Symposium on information, Computer, and Communications Security, 2009, pp. 183-194.

[20] L. Obrst, P. Chase, and R. Markeloff, "Developing an ontology of the cyber security domain," Proceedings of Semantic Technologies for Intelligence, Defense, and Security (STIDS), pp. 49-56, 2012.

[21] S. E. Parkin, A. van Moorsel, and R. Coles, "An information security ontology incorporating human-behavioural implications," in Proceedings of the 2nd International Conference on Security of Information and Networks, 2009, pp. 46-55.

[22] M. Poveda-Villalón, M. C. Suárez-Figueroa, and A. Gómez-Pérez, "Validating ontologies with oops!," in *Knowledge Engineering and Knowledge Management*, ed: Springer, 2012, pp. 267-281.

[23] Aldo Gangemi. (2010). *Submissions:Sequence*. Available: http://ontologydesignpatterns.org/wiki/Submissions:Sequence

[24] G. Antoniou and F. Van Harmelen, "Web ontology language: Owl," in Handbook on ontologies, ed: Springer, 2004, pp. 67-92

Building an Ontology of Cyber Security

Alessandro Oltramari and Lorrie Faith Cranor
CyLab, Carnegie Mellon University
Pittsburgh, USA

Robert J. Walls and Patrick McDaniel
Department of Computer Science
Pennsylvania State University
University Park, USA

***Abstract*—Situation awareness depends on a reliable perception of the environment and comprehension of its semantic structures. In this respect, cyberspace presents a unique challenge to the situation awareness of users and analysts, since it is a unique combination of human and machine elements, whose complex interactions occur in a global communication network. Accordingly, we outline the underpinnings of an ontology of secure operations in cyberspace, presenting the ontology framework and providing two modeling examples. We make the case for adopting a rigorous semantic model of cyber security to overcome the current limits of the state of the art.**

Keywords— cyber security, ontology, situation awareness, ontology patterns.

I. Introduction

As disclosed by a recent report[1], there has been half a billion cyber security breaches in the first semester of 2014, matching the record set across the entire precedent year. In general, this alarming trend should not surprise when we consider that the bedrock of the Internet is a technological infrastructure built almost 35 years ago for trusted military communications and not for data exchange in the wild (see [1], p.58). The picture gets even worse when considering that the ability to grasp the risk and threats associated with computer networks is averagely poor: recent surveys have actually shown that 65% of the victims of intrusion and information theft in the private sector are notified by third parties and that the detection process usually takes up to 13 months (e.g., see [2], p.10).

Though not exhaustive, such rough statistics at least suggest that if the inadequacy of the technological infrastucture is a key aspect to explain the vulnerabilities of networked computer systems, the *human factor* also plays a central role. As proposed in [3], to improve situation awareness of users and security operators, a shift of focus from system to environment level is highly necessary when modeling cyber scenarios: to this end, a full-fledged science of cyber security needs to be founded, whose core tenet is *cognizing* the cyberspace as a hybrid framework of interaction between humans and computers, where security and privacy policies play a crucial role. As stated by [4], this *cognizance* depends on both a reliable perception of the elements of the environment and, most importantly for our work, on the explicit representation of their semantics. Accordingly, the current article presents the underpinnings of an ontology of secure cyber operations: by and large, the concepts and the relationships that structure this semantic model are peculiar to the domain. That is, notions that are suitable for representing security in the physical world cannot be directly transferred to the cyber environment (e.g., "attack attribution" [5]). We build upon existing ontologies, expanding them to support novel use cases as needed[2]. Our goal is to use the proposed ontology as basis for improving the situation awareness of cyber defenders, allowing them to make optimal operational decisions in every state of the environment.

The rest of the paper is organized as follows: Section II makes the case for the adoption of ontologies in the cyber security realm; Section III outlines the structure of 'CRATELO', a Three Level Ontology for the Cyber Security Research Alliance program funded by ARL[3], and describes two simple cyber scenarios modeled by means of our approach; finally, Section IV draws preliminary conclusions and outlines an agenda for future research.

II. Related Work

Every science is concerned with distinct objects and strives to build rigorous models of the phenomena involving them [6]: accordingly, the objects of a science of cyber security correspond to the attributes of (and the relations between) network of computer devices, security policies, and the tools and techniques of cyber attack and cyber defense [7]. Therefore, inasmuch as ontologies are formal models of a domain, building ontologies of the aforementioned attributes and relations is critical for the transformation of cyber security into 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 requirement 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" [8]. The need for controlled vocabularies and ontologies to make progress toward a science of cyber security is recognized in [9] and [10] as well. In this domain, ontologies would include the classification of cyber attacks, cyber incidents, and malicious and impacted software

[1] https://www.riskbasedsecurity.com/reports/2014-MidYearDataBreachQuickView.pdf

[2] For instance, exploiting material available in this portal: http://militaryontology.com/cyber-security-ontology.html

[3] http://cra.psu.edu/

programs. From our point of view, where the human component of cyber security is also essential, the analysis needs to be expanded to the different roles that attackers, users, defenders and policies play in the context of cyber security, the different tasks that the members of a team are assigned to by the team leader, and the knowledge, skills and abilities needed to fulfill them.

There has been little work on ontologies for cyber security and cyber warfare. Within a broader paper, there is a brief discussion of an ontology for DDoS attacks [11] and a general ontology for cyber warfare is discussed in [12]. To the best of our knowledge, Obrst and colleagues [13] provide the most comprehensive description of a cyber ontology architecture, whose vision has actually inspired the work presented in this paper (the scale of the project and its difficulties are also discussed by Dipert in [10]). By and large, efforts that have been made toward developing ontologies of cyber security, even when expressed in OWL, RDF or other XML-based formats, typically do not utilize existing military domain or middle-level ontologies such UCORE-SL[4]. With regard to human users and human computer interaction, the most important step in understanding a complex new domain involves producing accessible terminological definitions and classifications of entities and phenomena, as stressed in [9]. Discussions of cyber warfare and cyber security often begin with the difficulties created by misused terminology (such as characterizing cyber espionage as an attack): in this regard, the Joint Chiefs of Staff created a list of cyber term definitions that has been further developed and improved in a classified version[5]. None of these definitions, however, are structured as an ontology. Likewise, various agencies and corporations (NIST[6], MITRE[7], Verizon[8]) have formulated enumerations of types of malware, vulnerabilities, and exploitations. In particular MITRE, which has been very active in this field, maintains two dictionaries, namely CVE (Common Vulnerabilities and Exposures [9]) and CWE (Common Weakness Enumeration[10]), a classification of attack patterns (CAPEC - Common Attack Pattern Enumeration and Classification[11]), and an XML-structured language to represent cyber threat information (STIX - Structure Threat Information Expression[12]). Regardless of the essential value of these resources, without a "shared semantics" the sprawling definitions they contain are hard to maintain and port into machine-readable formats.

III. A Three-Level Ontology for the Cyber-Security Research Alliance

Top-level ontologies capture generic characteristics of world entities, such as spatial and temporal dimensions, morphology (e.g., parts, edges, sides), qualities (e.g., color, volume, electric charge), etc.; because of their inherent generality, they are not suited to model contextual aspects. Nevertheless, it's good practice to describe the fine-grained concepts that constitute a *domain-level* ontology in terms of foundational (or *top-level*) categories, adding core (or *middle-level*) notions to fill contingent conceptual gaps. For instance, an ontology of mineralogy should include notions like "basaltic rock", "texture" and "metamorphic reaction". In order to describe the meaning of those specific concepts, high-level categories such that "object", "quality" and "process" must be employed; the ontology should also define an intermediate notion like "metamorphism", which is common across domains (biology, chemistry, computer science, architecture, etc.), to explain how the different phases, end products, and features of metamorphic reactions are bound together.

Our ontology of cyber security makes no exceptions to the tripartite layering described above: in particular, CRATELO is an ontological framework constituted of a domain ontology of cyber operations (OSCO), designed on the basis of DOLCE top ontology extended with a security-related middle-level ontology (SECCO). The three levels of CRATELO (schematized in figure 1) currently include 223 classes and 131 relationships (divided into 116 object properties and 15 datatype properties) and encoded in OWL-DL. The expressivity of the ontology is SRIQ, a decidable extension of the description logic SHIN (see [14] for more details).

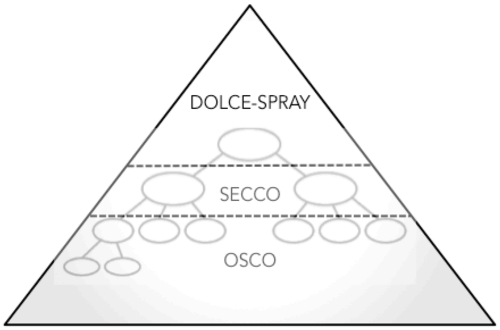


Figure 1: The schematics of CRATELO

A. *Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE)*

DOLCE is part of a library of foundational ontologies for the Semantic Web developed under the WonderWeb EU project[13]. As reflected in the acronym, DOLCE holds a cognitive bias, i.e., aiming at capturing the conceptual primitives underlying natural language and commonsense reasoning [15]. In order to reduce the complexity of the axiomatisation, in the current work we adopt DOLCE-SPRAY[14], a simplified version of DOLCE [16].

The root of the hierarchy of DOLCE-SPRAY is ENTITY, which is defined as the class of anything that is identifiable as an object of experience or thought. The first relevant distinction is among CONCRETE ENTITY, i.e., whose instances

[4] http://www.slideshare.net/BarrySmith3/universal-core-semantic-layer-ucoresl
[5] http://publicintelligence.net/dod-joint-cyber-terms/
[6] http://www.nist.gov/
[7] http://www.mitre.org/
[8] http://www.verizon.com/
[9] https://cve.mitre.org/
[10] http://cwe.mitre.org/
[11] https://capec.mitre.org/
[12] https://stix.mitre.org/language/version1.1.1/
[13] http://wonderweb.man.ac.uk/
[14] Categories are indicated in small caps; relationships in italics. Mutiple individuals instantiating the same category are denoted by adding an 's' to the category name (e.g., REQUIREMENTs). Presenting the axiomatisation of DOLCE-SPRAY is out of scope in this paper.

- CONCRETE ENTITY
  - CONTINUANT
    - AGENT
      - PERSON
      - GROUP
        - SOCIAL GROUP
    - OBJECT
      - ARTIFACT
      - NATURAL ENTITY
    - SUBSTANCE
  - PHYSICAL QUALITY
    - TEMPORAL LOCATION
    - SPATIAL LOCATION
    - COMPOSITE QUALITY
  - OCCURRENT
    - PROCESS
    - ACTION
    - STATE
- ABSTRACT ENTITY
  - ABSTRACT QUALITY
  - INFORMATION
  - CHARACTERIZATION
    - ROLE
    - PLAN
      - POLICY
    - TASK
    - REQUIREMENT

Figure 2: DOLCE-SPRAY backbone taxonomy

are located in definite spatiotemporal regions, and ABSTRACT ENTITY, whose instances don't have inherent spatiotemporal dimensions. CONCRETE ENTITY is further divided into CONTINUANT, OCCURRENT, and QUALITY, respectively entities with inherent spatial parts (e.g., artifacts, animals, substances), entities with inherent temporal parts (e.g., events, actions, states) and entities whose existence depends on their host (for instance 'the color of a flower', 'the duration of a football game', 'the area of a construction site', etc.). DOLCE's basic ontological distinctions are maintained in DOLCE-SPRAY: the substantial differences come from a) merging ABSTRACT and NON–PHYSICAL–ENDURANT categories into DOLCE-SPRAY's ABSTRACT ENTITY and b) by breaking the class QUALITY into PHYSICAL QUALITY and ABSTRACT QUALITY, moving the latter under the branch ABSTRACT ENTITY. Accordingly, the class ABSTRACT QUALITY designates the qualities that don't have any defining spatiotemporal dimension, such as the price of goods, the usefulness of a service, etc. A sibling of ABSTRACT QUALITY under the ABSTRACT ENTITY branch, INFORMATION refers to any content that can be conveyed by some physical OBJECT, from the metal boards used for road signs to the memory location of a Python script. CHARACTERIZATION is defined as a mapping of *n*-uples of individuals to truth-values. Individuals belonging to CHARACTERIZATION can be regarded to as 'reified concepts' (e.g., 'manufactured object'), and the irreflexive, antisymmetric relation *characterizes* associates them with the objects they denote ('a collection of vintage shoes'). Among the relevant sub-types of CHARACTERIZATION we can find: ROLE, i.e., the classification of an entity according to a given context or perspective (e.g., 'instructor'); PLAN, namely the generic description of an action (such as 'the disassembly of a 9mm'); TASK, that is a representation of the specific steps that are needed to execute an ACTION according to a PLAN (e.g., 'removing the magazine', 'pull back the slide'); REQUIREMENT, whose instances can be seen as the conditions that need to be satisfied as part of a PLAN (e.g., 'the weapon must be clear before proceeding'). A specific sub-class of PLAN is POLICY, whose instances need to satisfy specific REQUIREMENTs adopted or proposed by some SOCIAL GROUP (e.g., a government, a party, a no profit association, a private company, etc.). In general, the branch of DOLCE-SPRAY rooted on CHARACTERIZATION distills the extensions introduced in [17]. An overview of DOLCE-SPRAY backbone taxonomy is represented in Figure 2.

B. Security Core Ontology (SECCO)

This section outlines a set of security concepts based on DOLCE-SPRAY primitives.

An entity is a THREAT ϕ for an ASSET α *valued by* a STAKEHOLDER σ and *protected by* a DEFENDER δ, if and only if ϕ is *used by* an ATTACKER κ to *exploit* a VULNERABILITY ϖ of α in an OFFENSIVE_OPERATION τo. To prevent τo, a specific collection of SECURITY_REQUIREMENTS υs need to *be satisfied by* a SECURITY_POLICY π, enforced to protect α. But if τo strikes, δ has to promptly defend α, performing a suitable DEFENSIVE_OPERATION δo to *deploy* a COUNTERMEASURE χ for neutralizing PAYLOAD ψ conveyed by τo[15]. The class OPERATION can be represented as the union of τo and δo: any OPERATION o is carried out on the basis of a MISSION-PLAN λ whose sequence of MISSION_TASKS ξs *are executed in* o[16]. Note that in order to delineate λ in a DEFENSIVE_OPERATION δo, δ would also need to run a RISK-ASSESSMENT μ of the RISK ρ *associated to* ξs (datatype properties can be used to represent ρ as a parameterization of the expected losses, probabilities of attack, etc.)[17]. The formalization below (1-30) represents a basic alignment between SECCO and DOLCE-SPRAY. The relations *isPartOf*, *participates* (and its inverse *hasParticipant*), *isQualityOf*, *characterizes*, *definedIn*, *satisfies hasRole*, *hasRequirement*, are imported from DOLCE-SPRAY. We used self-explanatory abbreviations (e.g., OFF_OP instead of OFFENSIVE_OPERATION) to keep the list compact, when possible. For reasons of space, presenting a comprehensive set of axioms for SECCO is out of scope in this paper.

$$\text{ATTACKER}^{18} \sqsubseteq \text{ROLE} \sqcap \forall \mathit{characterizes}.\text{AGENT} \tag{1}$$

$$\text{DEFENDER} \sqsubseteq \text{ROLE} \sqcap \forall \mathit{characterizes}.\text{AGENT} \tag{2}$$

[15] Both countermeasures and payloads are artifacts of some sort, e.g., an antidote and a poison.

[16] o can be a single ACTION or a complex collection of interconnected actions.

[17] Although risk assessment needs to be done preemptively, continuous monitoring is also required for up-to-date situational awareness.

[18] In our model, instances of ATTACKER, DEFENDER and STAKEHOLDER are not equal to instances of PERSON, GROUP and, in general, AGENT. In this perspective, 'Alessandro' (instance of PERSON) *qua* DEFENDER would correspond to team member 'Alpha1' (instance of DEFENDER). *Qua*-entities have been formally analyzed in [33]. Also, since in different situations a defender may play the role of an attacker (and vice versa), we don't consider the two classes as disjoint.

$$\mathrm{STAKEHOLDER} \sqsubseteq \mathrm{ROLE} \sqcap \forall \mathit{characterizes}.\mathrm{AGENT} \tag{3}$$

$$\mathrm{STAKEHOLDER} \sqsubseteq \neg (\mathrm{ATTACKER} \sqcup \mathrm{DEFENDER})\text{[19]} \tag{4}$$

$$\mathrm{ASSET} \sqsubseteq \mathrm{ROLE} \sqcap \forall \mathit{characterizes}\,(\mathrm{OBJECT} \sqcup \mathrm{INFORMATION}) \tag{5}$$

$$\mathrm{ASSET} \sqsubseteq \neg\ \mathrm{THREAT} \tag{6}$$

$$\mathrm{THREAT} \sqsubseteq \mathrm{ROLE} \sqcap \forall \mathit{characterizes}\,(\mathrm{OBJECT} \sqcup \mathrm{INFORMATION}) \tag{7}$$

$$\mathrm{THREAT} \sqsubseteq \neg\ \mathrm{ASSET} \tag{8}$$

$$\mathrm{SEC\_REQ} \sqsubseteq \mathrm{DEF\_REQ} \sqsubseteq \mathrm{REQUIREMENT} \tag{9}$$

$$\mathrm{SECURITY\_POLICY} \sqsubseteq \mathrm{POLICY} \sqcap \forall \mathit{satisfies}.\mathrm{SEC\_REQ} \tag{10}$$

$$\mathrm{OFF\_REQ} \sqsubseteq \mathrm{REQUIREMENT} \tag{11}$$

$$\mathrm{OFF\_REQ} \sqsubseteq \neg\ \mathrm{DEF\_REQ} \tag{12}$$

$$\mathrm{DEF\_REQ} \sqsubseteq \neg\ \mathrm{OFF\_REQ} \tag{13}$$

$$\mathrm{OPERATION} \sqsubseteq \mathrm{ACTION} \tag{14}$$

$$\mathrm{DEF\_OP} \sqsubseteq \mathrm{OPERATION} \tag{15}$$

$$\mathrm{OFF\_OP} \sqsubseteq \mathrm{OPERATION} \tag{16}$$

$$\mathrm{OFF\_OP} \sqsubseteq \neg\ \mathrm{DEF\_OP} \tag{17}$$

$$\mathrm{DEF\_OP} \sqsubseteq \neg\ \mathrm{OFF\_OP} \tag{18}$$

$$\mathrm{MISSION\_PLAN} \sqsubseteq \mathrm{PLAN} \tag{19}$$

$$\mathrm{MISSION\_TASK} \sqsubseteq \mathrm{TASK} \sqcap \forall \mathit{isDefinedin}.\mathrm{MISSION\_\_PLAN} \tag{20}$$

$$\mathrm{RISK} \sqsubseteq \mathrm{ABST\_QUALITY} \sqcap \forall \mathit{isQualityOf}.\mathrm{MISSION\_\_TASK} \tag{21}$$

$$\mathrm{RISK\_ASSESSMENT} \sqsubseteq \mathrm{ACTION} \sqcap \exists \mathit{hasParticipant}.\mathrm{RISK} \tag{22}$$

$$\mathrm{COUNTERMEASURE} \sqsubseteq \mathrm{ARTIFACT} \sqcap \forall \mathit{participates}.\mathrm{DEF\_OP} \tag{23}$$

$$\mathrm{PAYLOAD} \sqsubseteq \mathrm{ARTIFACT} \sqcap \forall \mathit{participates}.\mathrm{OFF\_OP} \tag{24}$$

$$\mathrm{VULNERABILITY} \sqsubseteq \mathrm{ABST\_QUALITY} \sqcap \forall \mathit{isQualityOf}.\mathrm{ASSET} \tag{25}$$

$$\begin{aligned}\mathrm{DEF\_OP} \equiv\ & \exists \mathit{hasParticipant}.\mathrm{DEFENDER} \\ & \sqcap \exists \mathit{executes}.\mathrm{MISSION\_PLAN} \\ & \sqcap \exists \mathit{hasParticipant}.\mathrm{COUNTERMEASURE} \\ & \sqcap \exists \mathit{hasRequirement}.\mathrm{DEF\_REQ}\end{aligned} \tag{26}$$

$$\begin{aligned}\mathrm{OFF\_OP} \equiv\ & \exists \mathit{hasParticipant}.\mathrm{ATTACKER} \\ & \sqcap \exists \mathit{executes}.\mathrm{MISSION\_PLAN} \\ & \sqcap \exists \mathit{hasParticipant}.\mathrm{PAYLOAD} \\ & \sqcap \exists \mathit{hasRequirement}.\mathrm{OFF\_REQ}\end{aligned} \tag{27}$$

$$\mathrm{ATTACKER} \equiv \forall \mathit{exploits}.\mathrm{VULNERABILITY} \sqcap \exists \mathit{uses}.\mathrm{THREAT} \tag{28}$$

$$\mathrm{DEFENDER} \equiv \forall \mathit{protects}.\mathrm{ASSET} \sqcap \exists \mathit{uses}.\mathrm{COUNTERMEASURE} \tag{29}$$

$$\mathrm{STAKEHOLDER} \equiv \forall \mathit{values}.\mathrm{ASSET} \sqcap \exists \mathit{enforces}.\mathrm{SECURITY\_POLICY} \tag{30}$$

SECCO's categories are positioned at a too coarse-level of granularity to capture the details of domain-specific scenarios: properties like THREAT, VULNERABILITY, ATTACK, COUNTERMEASURE, ASSET are orthogonal to different domains and, in virtue of this, they can be predicated of a broad spectrum of things: for instance, infections are a threat to the human body, Stuxnet is a threat to PLCs, the impact of large asteroids on the Earth's surface is a threat to the survival of organic life forms, dictatorship is a threat to civil liberties, and so on and so forth. Though there seems to be a consensus in the literature on the core ontological concepts of security (see [18] and [19]), the minimal set presented here has been occasionally expanded along alternate directions. For instance, Fenz and Ekelhart [20] introduce the concept of 'control', by means of which stakeholders implement suitable countermeasures to mitigate known vulnerabilities of assets[20]. A 'policy', in this context, is defined as a regulatory or organizational form of control (SECCO definition of POLICY is more functionality-centered). Fenz and Ekelhart [20] also outline a taxonomy of assets, distinguishing 'tangible' (e.g., 'wallet') from 'intangible' ones (e.g., 'credit card credentials'), where the former can be furthermore split into 'movable' (e.g., 'car', 'jewelry') and 'unmovable' (e.g., 'house', 'land'). Interestingly enough, Fenz and Ekelhart reify the procedure of assessing a risk into the concept of 'rating', whose attributes can be expressed qualitatively (e.g., in Likert scale – high, medium and low) or quantitatively (measuring the probability of a risk). Avižienis and colleagues present a comprehensive analysis of security where the notion of 'fault' is introduced to denote an interruption of the services delivered by a given system in the environment [21]. A middle-level ontology of security can be possibly extended beyond SECCO: in this respect, the key contribution of this module doesn't rely on the coverage (or 'concept density' – see [22], p. 187) of security primitives but on the formalization driven by a top-level ontology. Our approach has some similarities with the effort described in [23], though Massacci and colleagues were principally concerned with the ontological analysis of a specific software development methodology, Secure Tropos.

C. Ontologies of Secure Cyber Operations (OSCO)

One of the major cyber security problems for government and corporations is the widespread "operational chaos" experienced by analysts, as Michael Susong has recently called the phenomenon of "having too many alarms (false positives) in a network, not enough trained people to deal with them, and a consequent poor prioritization of risks and countermeasures"[21]. In this regard, the objective of an ontology of cyber security is to shape that chaos into a framework of meaningful and reusable chunks of knowledge, turning the operational disarray into a systematic model by means of which cyber analysts can improve their situation awareness. As mentioned in section 1, the key to this augmented *cognizance* relies on a consistent assessment of the context and on a comprehensive understanding of its elements at the semantic level. But how is a cyber operation usually defined? In a document released in 2010, the Joint Chiefs of Staff describes a "cyberspace operation" as the "employment of cyber capabilities where the primary purpose is to achieve objectives in or through cyberspace. Such operations include computer network operations and activities to operate and defend the Global Information Grid" [24]. Drawing on this broad definition and relying on DOLCE-SPRAY and SECCO, in OSCO we represent a CYBER_OPERATION ψ as an OPERATION *executed by* a CYBER_OPERATOR φ , who can play either the role of DEFENDER in a DEFENSIVE_CYBER-OPERATION or the role of ATTACKER in an OFFENSIVE_CYBER-OPERATION. In the context of cyber security we can also distinguish between those OFFENSIVE_CYBER_OPERATIONS whose MISSION-PLANS satisfy the OFFENSIVE_REQUIREMENT of remaining undetected, and those that don't: we use the class CYBER_EXPLOITATION to the denote the former, and CYBER-ATTACK for the latter. As Lin points out in [5], from a technical viewpoint cyber-attacks and cyber exploitations are very similar: they use the same access paths and focus on the same vulnerabilities. The difference is on the delivery and

[19] Note that δ and σ may or may not coincide: in the second case, the latter needs to delegate the former to act in her behalf. The notion of delegation (and trust) in agent ontologies has been extensively studied by [26], but it's currently not included in CRATELO, as (6) shows.

[20] In cyber security, exploitations of unknown vulnerabilities correspond to the so-called Zero-Day Attacks.

[21] Dr. Micheal Susong is an Intelligence Subject Matter Expert affliated to iSIGHT Partners; he gave an invited talk at Carnegie Mellon University on September 8th, 2014.

execution of the PAYLOAD that must be performed undetectably in CYBER_EXPLOITATIONs (e.g., port scanning or SQL injections). The list of class-inclusions below (33-51) denotes the alignment between OSCO and SECCO categories and some specializations of OSCO domain concepts. For reasons of space we could not include a formal characterization of specific cyber threats and cyber vulnerabilities (comprehensive classifications can be consistently found in military reports, doctrines and academic articles - see [25] [26] [27]).

$$\text{CYBER\_OPERATION} \sqsubseteq \text{OPERATION} \tag{31}$$

$$\text{OFF\_CYBER\_OP} \sqsubseteq \text{CYBER\_OPERATION} \tag{32}$$

$$\text{DEF\_CYBER\_OP} \sqsubseteq \text{CYBER\_OPERATION} \tag{33}$$

$$\text{OFF\_CYBER\_OP} \sqsubseteq \text{OFF\_OP} \tag{34}$$

$$\text{OFF\_CYBER\_REQ} \sqsubseteq \text{OFF\_REQ} \tag{35}$$

$$\text{DEF\_CYBER\_REQ.} \sqsubseteq \text{DEF\_REQ} \tag{36}$$

$$\text{UNDETECTABILITY} \sqsubseteq \text{OFF\_CYBER\_REQ} \tag{37}$$

$$\text{CYBER\_COUNTERMEASURE} \sqsubseteq \text{COUNTERMEASURE} \tag{38}$$

$$\text{CYBER\_ASSET} \sqsubseteq \text{ASSET} \tag{39}$$

$$\text{CYBER\_THREAT} \sqsubseteq \text{THREAT} \tag{40}$$

$$\text{CYBER\_SEC\_REQUIREMENT} \sqsubseteq \text{SEC\_REQUIREMENT} \tag{41}$$

$$\text{CYBER\_SECURITY\_POLICY} \sqsubseteq \text{SECURITY\_POLICY} \tag{42}$$

$$\text{CYBER\_VULNERABILITY} \sqsubseteq \text{VULNERABILITY} \tag{43}$$

$$\begin{aligned}\text{CYBER\_ATTACKER} \sqsubseteq\ & \text{ATTACKER}\\ & \sqcap \forall\, \textit{exploits}.\text{CYBER\_VULNERABILITY}\\ & \sqcap \exists uses.\, \text{CYBER\_THREAT}\end{aligned} \tag{44}$$

$$\begin{aligned}\text{CYBER\_ANALYST} \sqsubseteq\ & \text{DEFENDER}\\ & \sqcap \forall\, \textit{protects}.\text{CYBER\_ASSET}\\ & \sqcap \exists uses.\, \text{CYBER\_COUNTERMEASURE}\end{aligned} \tag{45}$$

$$\begin{aligned}\text{CYBER\_STAKEHOLDER} \sqsubseteq\ & \text{STAKEHOLDER}\\ & \sqcap \forall\, \textit{values}.\text{CYBER\_ASSET}\\ & \sqcap \exists enforces.\, \text{CYBER\_SECURITY\_POLICY}\end{aligned} \tag{46}$$

$$\begin{aligned}\text{CYBER\_ATTACK} \sqsubseteq\ & \text{OFF\_CYBER\_OP}\\ & \sqcap \exists hasParticipant.\, \text{CYBER\_ATTACKER}\\ & \sqcap \neg\exists hasRequirement.\, \text{UNDETECTABILITY}\end{aligned} \tag{47}$$

$$\begin{aligned}\text{CYBER\_EXPLOITATION} \sqsubseteq\ & \text{OFF\_CYBER\_OP}\\ & \sqcap \exists hasParticipant.\, \text{CYBER\_ATTACKER}\\ & \sqcap \exists hasRequirement.\, \text{UNDETECTABILITY}\end{aligned} \tag{48}$$

$$\begin{aligned}\text{DEF\_CYBER\_OP} \sqsubseteq\ & \text{DEF\_OP}\\ & \sqcap \exists hasParticipant.\, \text{CYBER\_ANALYST}\\ & \sqcap \exists hasRequirement.\, \text{DEF\_CYBER\_REQ}\end{aligned} \tag{49}$$

Since the development of a full-scale domain ontology is currently underway within our project, for the sake of this article we will limit ourselves to model only two sample scenarios.

1) Example 1:RETRIEVE_FILE_SECURELY

Figure 3 represents CRATELO's classes and relationships used to model the *Retrieve File Securely* scenario. For issues of visualization, the diagram covers only the most salient notions involved in this cyber operation. In order to retrieve a file without exposing a computer system – and possibly an entire network – to cyber threats, some specific security requirements need to be fulfilled while carrying out that operation. In particular, as it is also the case for other kinds of CYBER-OPERATION, RETRIEVE-FILE-SECURELY must occur over a secure channel of a network, from authenticated computer(s) and through authorized server(s). By and large, abiding to these security requirements while executing the mission-tasks should lead to mission accomplishment. The composite RETRIEVE-FILE-SECURELY-TASK can be further divided into simpler temporally-structured and logically-connected subtasks. Accordingly, a request for a file can be sent to an authenticated server only after locating the desired file in the network; the inspection of the file can trivially occur only once the file has been obtained; and so on and so forth. In CRATELO we can express these basic temporal constraints by means of the foundational layer: in fact, DOLCE includes an adaptation of Allen's axioms [28], which are considered as a powerful logical theory for temporal representation and reasoning (the formalization of these axioms has also been maintained in DOLCE-SPRAY). Moreover, if malware is detected, the file must be removed from the host: the deployment of this preventive countermeasure aims at avoiding a disruption of the isolated computer node and a cyber attack to the network it belongs to. This countermeasure can be expressed as a conditional rule formalized in CRATELO by using an additional modeling apparatus, i.e., the Semantic Web Rule Language (SWRL)[22], which extends OWL-DL axioms. By including rule-based mechanisms in CRATELO we also comply with the core requisites described in [13] of a full-fledged cyber ontology architecture.

As the example exposes, one of the key design principles underlying CRATELO is to separate the temporal dynamics of cyber operations from the abstract generalizations used to describe them, i.e., plans, tasks, requirements. This approach consents to model a cyber operation as an ontology pattern grounded on the top level dyad ACTION-CHARACTERIZATION, unfolded by the middle-level tetrad OPERATION-MISSION_PLAN-MISSION_TASK-SEC_REQUIREMENT, and specified by CYBER_OPERATION-CYBER_MISSION_PLAN-CYBER_MISSION_TASK-CYBER_SECURITY_REQUIREMENT. In recent years, 'ontology patterns' have become an important instrument for conceptual modeling [29]: the rationale, as our work suggests, is to identify some minimal knowledge structures within an ontology to be used for modeling a problem (in this regard, the ontology remains the reference framework whereby the pattern can be expanded). This methodology is also ideal from a reasoning standpoint. For instance, in [30] the authors state that "mission activities are tasks focused on answering mission questions" (where the latter can be seen as partially overlapping the notion of security requirement): but an ontology that fails to discriminate 'activities' from 'tasks' would likely be affected in its inference capabilities, in the degree that reasoning over tasks that have not been executed yet – i.e., that are not activities – would not be supported. It's not difficult to imagine the circumstances where this limit can become a serious drawback for a cyber analyst: mental simulation is commonly adopted by humans to foresee the outcomes of an action before performing it [31], and a semantic framework where mission activities and tasks are conceptually viewed as the same entity precludes that, and might eventually result into pervasive logical inconsistencies (if the ambiguity is not somehow reduced). On the contrary, an ontology-pattern based on CRATELO allows to specify cyber operations at a sufficient level of conceptual granularity.

[22] http://www.w3.org/Submission/SWRL/

2) Example 2: INTRUSION_DETECTION

In a simplified scenario where an SQL injection attack is launched, a defensive cyber operation of INTRUSION_DETECTION can be divided into three essential sub-actions (and corresponding tasks): 1) block the IP address of the attacker; 2) to escalate the level of response; 3) to block all external connections and 4) redirect the incoming traffic to a honeypot for further inspection. Who can perform these actions? In the real world, cyber analysts with different responsibilities and privileges usually form a response team: for instance, we can indicate with L1, L2 and L3 the incremental levels of expertise of cyber analysts. Accordingly, 1) would only be performed by L1 analysts; 2) can only be performed by L1 analysts toward L2 analysts or by L2 toward L3; 3) can only be executed by L2 analysts and 4) only by L3. As a matter of fact, gauging which action fits better the situation is not a one-shot decision, but rather a multi-stage evaluation process where the situational awareness of cyber analysts frequently changes Also, each of those sub-actions has incremental costs and inversely proportional risks: for instance, if blocking all the connections to a web server eliminates the risks of a reiterated attack, suspending the network traffic has a severe impact on the system functionality (e.g., no data access for authorized third parties): escalation, in this context, is an effective means to prevent risk mismanagement. Although this simplified scenario gives only a partial account of the actions that actual analysts have at their disposal, using an ontology of cyber security like CRATELO to model intrusion detection can clearly represent a mean to improve situational awareness and fill the *semantic gap* [32] in our understanding of the cognitive demands in the cyber world. Figure 4 presents a partial view of CRATELO categories and relations used for intrusion detection.

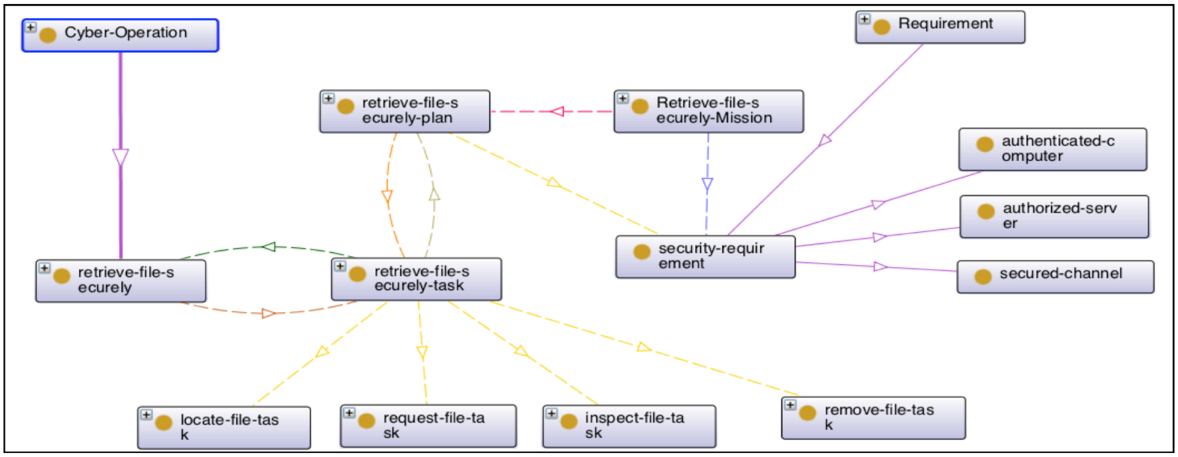


Figure 3 – A visualization of the RETRIEVE-FILE-SECURELY cyber operation modeled in CRATELO. Legend of the arc types: 'has subclass' (purple); 'is executed in' (green); 'executes' (brown); 'has part' (yellow); 'defines task' (orange); 'is defined in task' (ochre); 'satisfies (all)' (fuchsia); 'satisfies (some)' (electric blue).

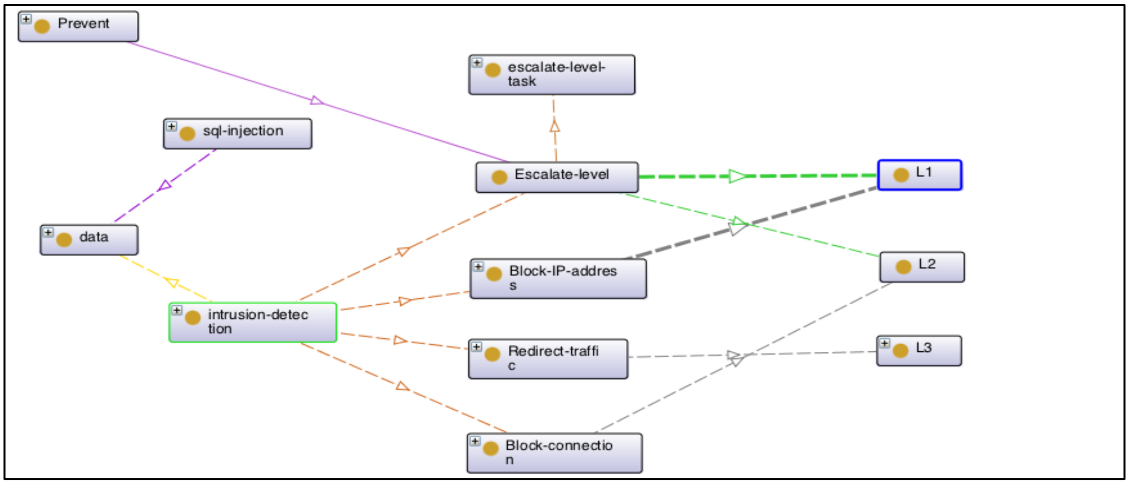


Figure 4 – A subset of actions that can be performed in a cyber operation of INTRUSION_DETECTION. This diagram shows some of the interdependencies between classes of actions and levels of expertise of cyber analysts. Legend of the arc types: 'has subclass' (solid purple); 'targets' (dotted purple); 'defend' (yellow); 'has part' (brown); 'executes task' (light brown); 'involves (only) agent' (gray); 'involves (only disjunction)' (green).[23]

[23] Figure 3-4 were generated and exported using Ontograf (http://protegewiki.stanford.edu/wiki/OntoGraf), a visualization plug-in for Protégé. Even within the same ontology, Ontograf automatically assigns different colors to arcs when a new figure is created: this explains mismatch of colors between the two figures.

IV. Conclusions And Future Work

Notwithstanding the proliferation of taxonomies, dictionaries, glossaries, and terminologies of the cyber landscape, building a comprehensive model of this domain remains a major objective for the community of reference, that includes government agencies, private organizations, researchers and intelligence professionals. There are multiple reasons behind the discrepancy between demand and supply of semantic models of cyber security. Although we cannot thoroughly address this topic here, we are firmly convinced that a great part of the problem is the lack of balance between the 'vertical' and the 'horizontal' directions of the effort. From one side, state of the art consists of several classifications of the domain, as argued in Section II: these efforts typically yield rich catalogs of cyber attacks, exploits and vulnerabilities. On the other side, a rigorous conceptual analysis of the entities and relationships that are encompassed by different cyber scenarios would also be needed, but little work has been done on this horizontal dimension (if we exclude the ongoing MITRE initiative described by Leo Obrst and colleagues in [13]). In this paper we placed ourselves on the second perspective: instead of presenting "yet another" catalog of cyber notions, an endeavor that remains however of undisputable relevance, we decided to explore in depth the semantic space of operations. Our investigation addresses cyber operations as complex entities where the human factor is as important as the technological spectrum: our ontological analysis is grounded on a bedrock of foundational concepts and reaches the domain of cyber operations through an intermediate layer where core notions are defined.

Future work will focus on the following research steps:

- extending SECCO with an ontology of risk;
- populating OSCO with a large set of cyber operations documented in the literature and learned from real-world case studies;
- designing and customizing a methodology for ontology validation based on "competency questions" submitted to domain experts (along to what has been proposed in [20]);
- running cyber warfare simulations within military exercises, collecting data to be modeled with CRATELO;
- studying ontology mappings between CRATELO and other semantic models (e.g., MITRE's Cyber Ontology Architecture), ensuring interoperability and reusability of the resource.

We are aware of the challenges ahead of us in pursuing this research agenda, which would usually be very difficult to implement. Nevertheless, we're also persuaded that, in the broad vision framed by the ARL Cyber Security Collaborative Research Alliance, what we have described illustrates a realistic work plan and a necessary step toward the foundation of a science of cyber security.

Acknowledgments

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

[1] Yannakogeorgos, P. and Lowther, A. B. "The Prospects of Cyber Deterrence: American Sponsorships of Global Norms," in *Conflict and Cooperation in Cyberspace*.: Taylor&Francis, 2013, pp. 49-77.

[2] L. Mattice, "Taming the "21st Century's Wild West" of Cyberspace?," in *Conflict and Cooperation in Cyberspace*.: Taylor&Francis, 2013, pp. 9-12.

[3] McDaniel, P., Rivera, B., Swami, A. "Toward a Science of Secure Environments," *Security and Privacy*, vol. 12, no. 4, pp. 68-70, July/August 2014.

[4] Endsley, M.R. "Toward a Theory of Situation Awareness in Dynamic Systems," *Human Factors*, vol. 37, no. 1, pp. 32-64, 1995.

[5] Lin, H. "Escalation Dynamics and Conflict Termination in Cyberspace," *Strategic Studies Quarterly*, vol. 6, no. 3, pp. 46-70, Fall 2012.

[6] Bunge, M. *Causality and Modern Science*. New York: Dover Publications, 1979.

[7] Kott, A."Towards Fundamental Science of Cyber Security," in *Network Science and Cybersecurity*, R. E. Pino, Ed. New York, 2014, vol. 55.

[8] The MITRE Corporation, "Science of Cyber-Security," The MITRE Corporation, McLean, VA, Technical 2010.

[9] Mundie, D. A. and McIntire, D. M. "The MAL: A Malware Analysis Lexicon," CERT® Program - Carnegie Mellon University , Technical 2013.

[10] Dipert, R. "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.

[11] Kotenko, I. "Agent-Based modeling and simulation of cyber-warfare between malefactors and security agents in internet ," in *19th European Conference on Modeling and Simulation*, 2005.

[12] D'Amico, A., Buchanan, L., Goodall, J. & Walczak, P. (2009) Mission impact of cyber events: Scenarios and ontology to express the relationship between cyber assets. [Online]. http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA517410

[13] Obrst, L., Chase, P., & Markeloff, R. "Developing an ontology of the cyber security domain," in *Seventh International Conference on Semantic Technologies for*

Intelligence, Defense, and Security, 2012, pp. 49-56).

[14] Horrocks, I., Kutz, O., Sattler, U. "The Irresistible SRIQ ," in *OWLED '05 - "OWL: Experiences and Directions"*, vol. 188, Galway, 2005.

[15] Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, Schneider, L. A. "The WonderWeb Library of Foundational Ontologies and the DOLCE ontology," Laboratory For Applied Ontology, ISTC-CNR, Technical Report 2002.

[16] Vetere G., Jezek E., Chiari I., Zanzotto F.M., Nissim M., Gangemi A. Oltramari A., "Senso Comune: A Collaborative Knowledge Resource for Italian," in *The People's Web Meets NLP: Collaboratively Constructed Language Resources.*: Springer Verlag, 2013, pp. 45-67.

[17] Gangemi, A., Mika, P. "Understanding the Semantic Web through Descriptions and Situations," in *On The Move to Meaningful Internet Systems - Lecture Notes in Computer Science*. Berlin-Heidelberg: Springer, 2003, vol. 2888, pp. 689-706.

[18] Salinesi, C., Wattiau, I., A. Souag, "Ontologies for Security Requirements: A Literature Survey and Classification," in *Advanced Information Systems Engineering Workshops*, vol. 112, 2012, pp. 61-69.

[19] Schumacher, M. "Toward a Security Core Ontology," in *Security Engineering with Patterns*. Berling-Heidelberg: Springer-Verlag, 2003, pp. 87-96.

[20] Fenz, S., Ekelhart, A. "Formalizing Information Security Knowledge," in *the International Symposium on Information, Computer, and Communications Security (ASIACCS '09)*, New York, pp. 183-194.

[21] Avižienis, A., Laprie, J., Randell, B., Landwehr, C. "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11-33, January-March 2004.

[22] Prévot, L., Borgo, S., Oltramari, A. "Interfacing Ontologies and Lexical Resources," in *Ontology and the Lexicon - A Natural Language Perspective*, C.R., Calzolari, N., Gangemi, A., Oltramari, A., Prévot, L. Huang, Ed. New York, USA: Cambridge University Press, 2010, pp. 185-200.

[23] Massacci, F., Mylopoulos, J., Paci, F., Thein, T.T., Yijun, Y. "An Extended Ontology for Security Requirements". In *CAiSE 2011 International Workshops*, vol. 83, London, 2011, pp. 622-636.

[24] Joint Staff Department of Defense. Joint Terminology for Cyber Operations. [Online]. http://afri.au.af.mil/cyber/Docs/panel1/Cyber_Lexicon.pdf

[25] Joint Chiefs of Staff, "The National Military Strategy for Cyberspace Operations," Department of Defense, 2006. [Online]. http://www.dod.mil/pubs/foi/joint_staff/jointStaff_jointOperations/07-F-2105doc1.pdf

[26] Air Force Doctrine Document, "Cyberspace Operations,".

[27] Simmons, C. B., Shiva, S. G., Bedi, H., Dasgupta "AVOIDIT: A Cyber Attack Taxonomy," in *9th Annual Symposium on Information Assurance (ASIA)*, Albany, NY, 2014, pp. 2-12.

[28] Allen, J.F. "An interval based representation of temporal knowledge," in *7th International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 1, Vancouver, 1983, pp. 221–226.

[29] Gangemi, A. and Presutti, V. "Ontology design patterns," in *Handbook on Ontologies*.: Springer , 2009, pp. 221-244.

[30] Morris, T.I., Mayron, L.M., Smith, W.B., Knepper, M.M., Reg, I., Fox, K.L. "A perceptually-relevant model-based cyber threat prediction method for enterprise mission assurance," in *IEEE Multi-disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support*, Miami Beach, 2011, pp. 60-65.

[31] Taylor, S.E., Pham L.B., Rivkin I.D., Armor D.A. "Harnessing the imagination. Mental simulation, self-regulation, and coping.," *American Psychologist* , vol. 53, no. 4, pp. 429-439, Apr 1998.

[32] Gonzalez, C., Ben-Asher, N., Oltramari, A., Lebiere, C. "Cognitive Models of Cyber Situation Awareness and Decision Making," in *Cyber Defense and Situational Awareness*, A., Wang, C., Erbacher, R. Kott, Ed.: Springer, 2014, vol. 62.

[33] Masolo, C., Guizzardi, G., Vieu, L., Bottazzi, E., Ferrario, R. "Relational Roles and Qua Individuals". In *AAAI Fall Symposium on Roles, an Interdisciplinary Perspective*, Virginia, USA. 2005.

An Incident Management Ontology

D. Mundie, R. Ruefle, A. Dorofee, J. McCloud, S. Perl, M. Collins
CERT®
Software Engineering Institute | Carnegie Mellon University
4500 Fifth Ave., Pittsburgh, PA, United States of America

***Abstract*—This paper outlines the need for and the development of an Incident Management Ontology. The Incident Management Ontology is derived from an Incident Management Meta-Model. We describe the shortcomings of the Incident Management Meta-Model and how the Incident Management Ontology addresses these shortcomings. The development of the Incident Management Ontology is outlined and the need for such an ontology is discussed. Related work is described and the Incident Management Ontology's potential uses and applications are presented.**

Keywords—Ontology, Incident Management, Description Logic

I. Introduction

When the JASON[1] Program within MITRE looked at the scientific community for ways to make cybersecurity "more scientific", their very first conclusion was that the security community needed "a common language and a set of basic concepts about which the security community can develop a shared understanding" [1], or in other words, a Cybersecurity ontology. The work described in this report is part of an ongoing effort within CERT® to build such an ontology for incident management.

We believe that such formal models are the best way for the community to evolve towards a "science of cybersecurity", and that our incident management ontology can play a crucial role in improving incident management. The ontology's purpose is to create a common language for describing the processes and functions associated with CSIRTs. We intend to use the ontology to analyze existing CSIRTs, to define a standard set of processes and services that should be offered by CSIRT teams, to formalize roles and responsibilities, and to build an ontology based competency model for the knowledge, skills, and abilities required of team members.

This paper describes the evolution of our work on characterizing incident security teams from a natural-language text document to a formal ontology and analyzes the benefits that accrued in the process. When creating our ontology, we chose to use the W3C Ontology Web Language - OWL[2] due to its formalism and increasing use in the Semantic Web community. We feel this work may be a useful case study for others who are thinking about formalizing their own information security knowledge.

II. The Incident Management Meta-Model

In previous work [2], we aggregated a wide variety of incident management process models such as ISO 27002 [3] and NIST 800-61 [4]. From those sources we abstracted a generalized meta-model that captured the essential processes involved in incident management.

This meta-model was at the heart of what we previously called an Incident Management Body of Knowledge (IMBOK). It broke incident management activities into 18 high-level tasks organized by the incident management life cycle phases as Prepare, Protect, and Respond. It also included five non-procedural, crosscutting capabilities that constrain all the other tasks. The following outlines the phases and tasks and 5 crosscuts of the IMBOK:

A. The phases and tasks

1) Prepare

- Develop trusted relationships with external experts
- Provide staff with appropriate education and training
- Develop policies, processes, procedures
- Measure incident management performance
- Provide constituents with security education, training, and awareness
- Develop an incident response strategy and plan
- Improve defenses

2) Monitor and Detect

- Assist constituents with correcting problems identified by vulnerability assessment activities
- Detect and report events
- Monitor networks and information systems for security
- Perform risk assessments and vulnerability assessments on constituent systems

3) Respond

[1] "JASON is an independent scientific advisory group that provides consulting services to the U.S. government on matters of defense science and technology. [In 2010] JASON was asked by the Department of Defense to examine the theory and practice of cyber-security, and to evaluate whether there are underlying fundamental principles that would make it possible to adopt a more scientific approach." (http://fas.org/irp/agency/dod/jason/)

[2] OWL is based upon description logics. OWL supports those users who want the maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time). (http://www.w3.org/TR/owl-features/)

- Triage Incident
- Collect and preserve evidence
- Restore and validate the system
- Perform a postmortem review of incident management actions
- Integrate lessons learned with problem management process
- Analyze incident, including artifacts, causes, and correlations
- Determine and remove the cause of the incident

B. The 5 crosscuts

1) Manage information

2) Properly handle collected evidence following best practices

3) Manage the incident management team

4) Communicate incidents

5) Track and document incidents from initial detection through final resolution

C. Drawbacks to the Incident Management Meta-Model

Although the Incident Management Meta-Model provides a considerable simplification and consolidation of prior knowledge, it suffers from a number of drawbacks due to its knowledge representation formalism:

- The use of imperative verb forms expressing infinitive constructions means that each task is only partially represented, because the subject is implicit. This obfuscates, for example, the fact that some of the tasks (e.g. managing the team) are carried out by the team's managers, not by the incident responders.
- In general, the use of natural language makes machine processing of this knowledge representation difficult.
- In particular, there is no easy way to use this representation to perform modeling and simulation, nor to build applications on top of it.
- To keep the process model manageable, concepts have been abstracted to an unusable level, with no graceful way to expand them into a more detailed form. There is no way within this system, for example, to say what is meant by "defenses" in "improve defenses".
- Apart from including a glossary, this representation does not facilitate the use of a standardized vocabulary.
- Also to keep the process model manageable, related concepts have been combined, as in "restore and validate the system".
- Despite its relative compactness, this representation violates the "7 plus or minus 2" law [5] and is hard for users to take in at a glance and internalize.

III. From Meta-Model to Ontology

Recently we realized that many of the drawbacks of the IMBOK could be remediated by moving beyond the informal natural-language format of the body of knowledge, and instead building a formal ontology using OWL.

A. Ontologies

An ontology is simply a set of shared, precisely-defined concepts in a given domain, along with the relationships among those concepts. OWL (the Web Ontology Language) is a W3C recommendation that builds on earlier languages from DARPA and elsewhere [6], is a key component of the Semantic Web [7], and is currently the leading knowledge representation and reasoning language in computer science. OWL is descended from earlier attempts at usable knowledge representation systems such as expert systems, logical programming languages, frame-based reasoning systems, modal logic, KL-One [8], entity-relationship modeling, and the like [9]. Description Logics emerged as a flexible yet powerful knowledge representation tool as the relationships among these approaches were better understood and new ways to engineer logics and reasoning systems were discovered. Description Logics have been used projects ranging from the International Catalogue of Diseases [10] to Google's Knowledge Graph [11].

To build our IM ontology, we decomposed the 18 high-level tasks in the IMBOK meta-model into component concepts and their respective relationships.The concepts, also known as classes in the Description Logic community, are organized into a strict hierarchy of subclasses. The incident management tasks are composed of relationships among those classes. This separation of classes from relationships is the key to most modern knowledge formalisms, from KL-One [8] to OWL [12].

B. N-ary Relationships

The only relationships inherent in the Description Logic on which OWL is built are binary relationships consisting of two concepts (or objects) and a relationship between them. However, many of the relationships we want to model in incident management are "n-ary" relationships among more than just two objects. For example, training requires a relationship among at least three objects: the training itself, a trainer, and a trainee. There are a number of ways to handle this situation in OWL; for the IM ontology we used one of the techniques recommended by the W3C [13]. This technique consists of creating a new class that holds the relationships among the training concepts.

This requires a slight adjustment to our ways of thinking about relationships. To illustrate, the original meta-model tasks

(IM leaders) Develop trusted relationships with external experts.

(trainers) Provide staff with appropriate education and training.

become

developing external relationships:
involves external groups
produces trusted relationships
is performed by IM leaders

staff training:
is provided by either external or internal trainers
is provided to IM personnel

Once the reified relationships are in place, it becomes straightforward to enhance them with additional information. In full, these two classes actually are as follows in the ontology:

developing external relationships:
belongs to the prepare process
involves external groups
produces trusted relationships
is subject to the incident management crosscuts
is performed by IM leaders

staff training:
is provided by either external or internal trainers
is provided to IM personnel
is a training service
is part of the prepare process
is subject to the incident management crosscuts

The table In Appendix B gives a simple summary of the relationships in the ontology.

Figure 1 shows a screenshot of the IM ontology being edited in Protégé [14], the ontology development tool from Stanford that is widely used in the community. The display contains five panes giving five views of the ontology. The upper left pane shows the class hierarchy. The two most important classes are "activities" and "crosscuts". The activities are simply the tasks carried out by the incident management staff, while "crosscuts" or "principles" as Beebe and Clark call them [15] are pervasive constraints on the activities. In addition to those main classes, we needed eight auxiliary classes to describe the activities in full: incident components, IT components, knowledge assets, life cycle phases, organizational groups, quality standards, relationships, and team resources. These classes were identified using traditional ontology-mining techniques: we started with the terms in the meta-model, then clustered them and introduced class hierarchies based on our knowledge of the domain.

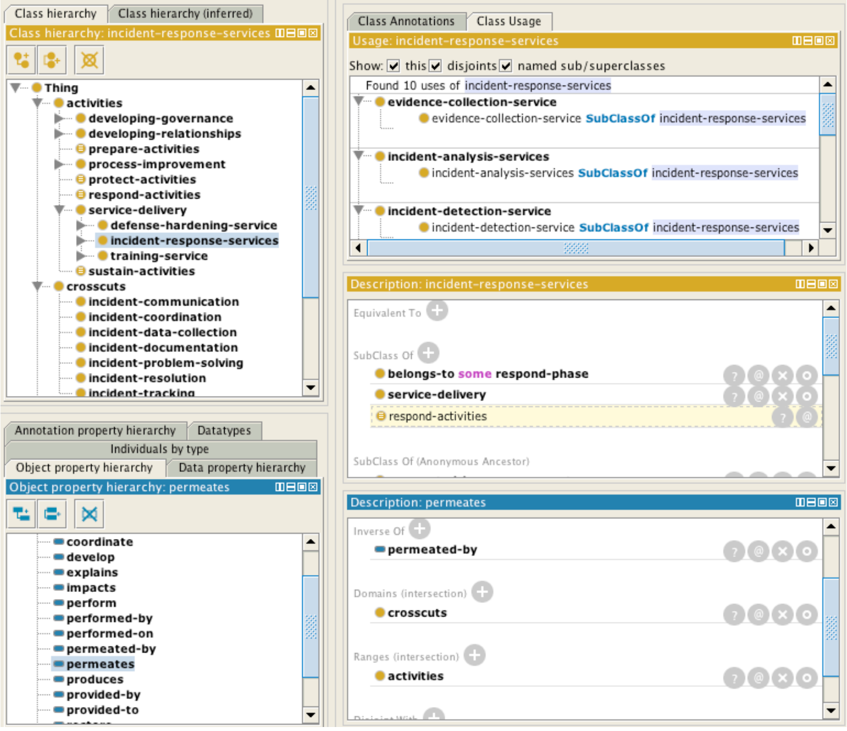


Figure 1 - The Incident Management Ontology Being Edited in Protégé. Note that the "crosscuts" class has grown and its members renamed during the development process.

The top two right-hand panes of the display show additional information about the selected class in the class hierarchy, in this case "incident response services". The top pane shows the usage of the selected class, while the second pane shows information about the class in terms of its subclasses, its superclasses, its members, any equivalent classes, and so forth.

The pane at the lower left of the screen shows the hierarchy of relationships, called "object properties" in

OWL. The "permeates" relationship has been selected. The lowest pane on the right describes that relationship in the ontology, showing that its domain is "CSIRT managers" and its range is "team resources", capturing the fact that CSIRT team managers acquire the team's resources.

C. *Overcoming the Drawbacks*

We believe that this formal IM ontology solves the problems noted in Section 2 for the IMBOK meta-model.

- The use of classes and relationships ensures that the knowledge is represented completely.
- This representation is machine-processable; Figure 2 shows a simple graphic automatically generated from the IM ontology using the OntoGraf tool [16] with a GraphViz post-processing script.
- The use of Description Logic (DL) ontologies for modeling and for constructing applications is well understood [9].
- The use of a strict class hierarchy gives us a user-friendly way to talk about concepts at any needed level of abstraction without complicating the IM ontology as a whole: we can talk about "security training", or "training", or "proactive services", or "incident management services", and the reasoning system will infer properties and type relationships as needed.
- The use of OWL annotations to capture definitions makes the IM ontology usable as a dictionary.
- Because of the class hierarchy and the formality of the system, there is no pressure to collapse concepts to keep the document small.
- Finally, the separation of entities from relationships reduces the complexity of the representation, and makes the structure of the IM ontology easier to absorb.

Figure 2 gives a high-level breakdown of the incident management activities. The "service delivery" activities are the most important, and Figure 2 expands that class to a further level of detail. Figure 3 shows a close-up of the root cause analysis environment, showing that it is performed by incident management personnel, that its goal is to explain root causes, that it is an incident analysis service, and so forth.

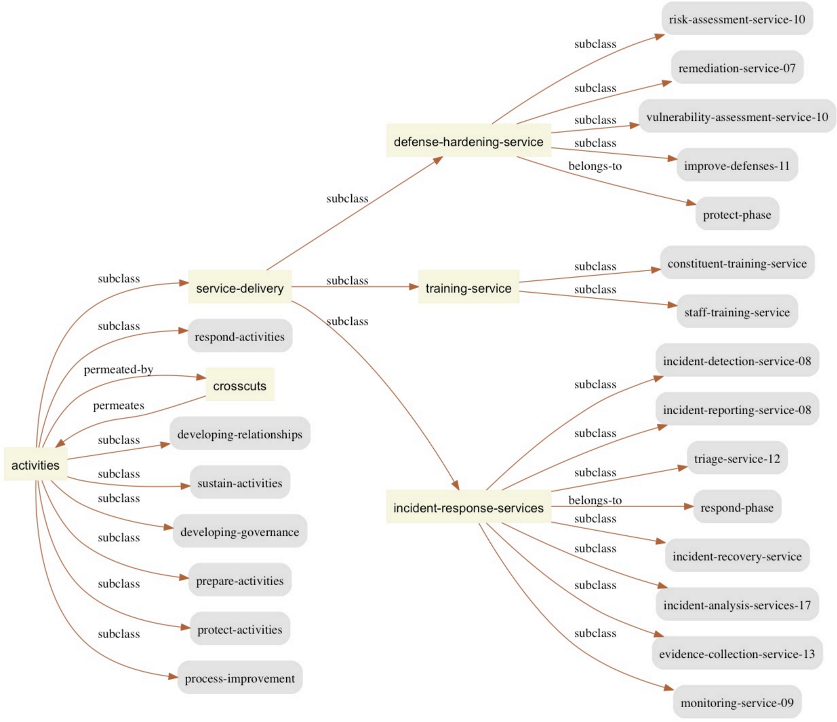


Figure 2 - The Activity Classes in the Ontology, with the Service-Delivery Activity Expanded

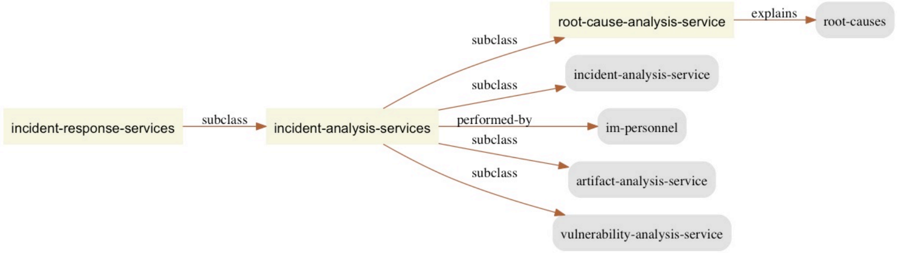

root-cause-analysis-service
explains
root-causes
subclass
incident-analysis-service
incident-response-services
incident-analysis-services
performed-by
im-personnel
artifact-analysis-service
vulnerability-analysis-service

Figure 3 - A Detailed View of Root Cause Analysis

D. *Other Benefits*

In addition to solving the difficulties we had with the Meta-Model, moving to a formal ontology had several other advantages.

1) Very flexible typing

We quickly grew fond of the ability to create new types simply by specifying the necessary and sufficient conditions for membership in the type. Earlier we had used a multidimensional organization system called facet maps [17] to achieve multiple categorizations for the Meta-Model, but class expressions are much more lightweight and flexible. They are like a very disciplined tagging system. To cite just one example, we realized at some point that although we want to retain the classification of activities by the life-cycle phase in which they are used (prepare, protect, detect, respond, etc.), there is no need to build the life-cycle phases into the class hierarchy. Instead we simply assert a "belongs-to" relationship between an activity and a life-cycle phase. Then we can define a "protect-activities" class where the membership condition is "an activity that belongs-to the protect phase" and the reasoner will automatically compute the members of the class.

2) More powerful Modeling

The n-ary relations that use binary relations to “reify” relations among individuals turned out to be a very effective method for packaging up domain knowledge in a taxonomic hierarchy. When it seemed clear that the different types of incident analysis were characterized by the goal of their analysis, it was trivial to add "explains" and "explained-by" relationships.

3) Improved knowledge visualization

A shortcoming of our Incident Management Meta-Model was the absence of a satisfactory visualization. After converting the Meta-Model into a formal ontology, we used OntoGraf [16] to export files in the GraphViz DOT format [18]. DOT is a text-based format that allows for customizable graphics.

E. *Individuals*

The real power of Description Logic ontology comes when an ontology is populated by individuals and reasoning is enabled. “Reasoning” is a key-functionality of semantic technologies and allows automatic inferences to be made using the rules and classes described by the ontology. The ability of OWL to be used at internet scale comes from the highly optimized and logically precise handling of both terminological, or *taxonomic*, knowledge in what the Description Logic community calls the TBox, and the contingent *assertional* knowledge about individuals in what the community calls the ABox [9].

We have not yet formally extended the Incident Management Ontology to real world individuals, but Figure 4 shows an example using two fictitious individual CSIRTs in the ontology. The Acme team, focused only on incident response, provides monitoring, incident detection, incident reporting, and incident analysis services. The National Team from Borduria focuses on vulnerability assessment, vulnerability analysis, incident analysis, performance measurement, and relationship building. As the diagram makes clear, the only service these two CSIRTs have in common is incident analysis.

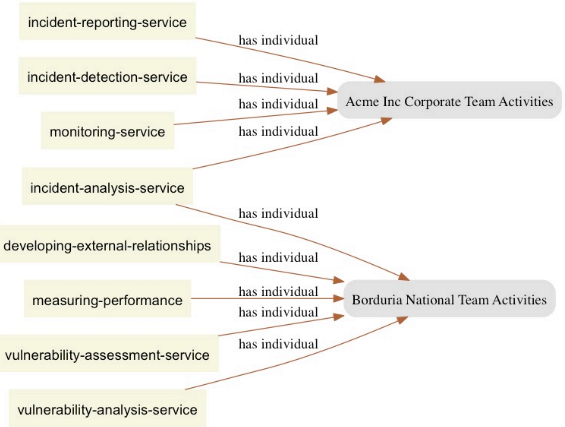

incident-reporting-service
incident-detection-service
monitoring-service
incident-analysis-service
developing-external-relationships
measuring-performance
vulnerability-assessment-service
vulnerability-analysis-service
has individual
Acme Inc Corporate Team Activities
Borduria National Team Activities

Figure 4 - A Comparison of Two Fictitious Incident Management Teams. Note that “has individual” denotes the membership of individuals in classes. Thus Acme is an individual member of the class of incident reporting services.

IV. RELATED RESEARCH

The seminal paper Formalizing Information Security Knowledge by Fenz et al. [19] presents the rationale for capturing information security best practices in an OWL ontology. Though it served as an influence for our ontology, the work addresses information security in general while our work focuses on incident management.

There have been many proposals for standardized incident handling process models; for a summary of the models that were used for our meta-model, see [2]. Although they incorporated much collective wisdom, none of them were based on a formal knowledge representation. Like our meta-model, the forensic framework of Beebe and Clark [15] aimed to assimilate existing practice into a comprehensive framework. The distributed, loosely-coupled incident response model of Millar, Osorno, and Reger [20] is a deeply-reasoned attempt to analyze and improve upon existing incident management practices based on scientific theory and simulation, but is not based on a formal ontology.

Furthermore, we found that many proposed ontologies that exist fail to capture all of the important relationships between members of organizations and the organizations themselves. These representations arise from an internal focus of an organization who has been victim to attack, and many ignore the roles and relationships between a CSIRT and incidents that occur.

Magklaras and Furnell [21] observe that incidents occur through misuse by individuals, but do not propose a formalized ontology of a taxonomy including this human-misuse concept. Classifications of individuals are made more distinguished based on behavior (e.g. accidental or intentional), and possible consequences of misuse correlated to these actions.

Wang and Guo’s [22] research in developing OVM (Ontology for Vulnerability Management) identifies individuals responsible for attacks, but the relationships amongst these individuals is not made clear. The formalizations within their work capture knowledge sufficient to answer questions about the assets targeted in an incident and mechanisms by which an incident takes place. While organization and individuals are clear in this work, further subdivisions of organizations and groups of individuals are not. No concept of trust appears in the ontology’s class hierarchy, making the risk of agents difficult to reason about. Chiang [23] proposed mapping the IT Security EBK [24] and ISO/IEC 27001 [25] standard to an incident ontology. The construct is similar to OVM, but has the benefit of subdivision of roles amongst individuals and groups. Subdivisions, however, are limited and the ontology will require additional, higher-level concepts to subsume various sibling classes of the hierarchy.

The most complete formalization framework in security that gathers all necessary information to incident management might be Ekelhart’s [26] move from simple security taxonomy to ontology. This work acknowledges the different threats and means for attacks, along with measurable reductions when safeguards are introduced. Even relationships amongst individuals in an organization and the roles they take are represented clearly. However, this research does not model subdivisions of an organization and the roles that multiple organizations can have (both within and in relation to one another). Different subdivisions of service types and measures of trust are not represented.

V. Next Steps

Future work on the Incident Management Ontology will focus on evaluting the ontology and using it to categorize incident response organizations. This work names CSIRT processes but does not yet describe them in full detail. Future work may include using existing standards, such as the Process Specific Language[27], to model the process flows for each service offered by a CSIRT in greater detail. We plan to evaluate the ontology by using it to analyze the processes performed by and services offered by incident response teams. A hypothesis we would like to test is whether there is a difference between the functions of CSIRTs and Coordination Centers. We are collecting data on both types of organizations and plan to analyze it using the ontology. We also plan to improve the ontology by adding axioms, more defined classes, and taking more advantage of reasoning capabilities.

VI. Acknowledgements

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. This material has been approved for public release and unlimited distribution. Carnegie Mellon® and CERT® are registered marks of Carnegie Mellon University. DM-0001433

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

References

1 McMorrow, D.: ‘Science of Cyber-Security’, in Editor (Ed.)^(Eds.): ‘Book Science of Cyber-Security’ (DTIC Document, 2010, edn.), pp.

2 Mundie, D.A., and Ruefle, R.: ‘Building an Incident Management Body of Knowledge’, in Editor (Ed.)^(Eds.): ‘Book Building an Incident Management Body of Knowledge’ (Citeseer, 2012, edn.), pp. 507-513

3 ISO, I., and Std, I.: ‘ISO 27002: 2005’, Information Technology-Security Techniques-Code of Practice for Information Security Management. ISO, 2005

4 NIST: ‘Special Publication 800-61, Revision 2’, Computer Security Incident Handling Guide, 2012, pp. 800-861

5 Miller, G.A.: ‘The magical number seven, plus or minus two: some limits on our capacity for processing information’, Psychological review, 1956, 63, (2), pp. 81

6 Motik, B., Patel-Schneider, P.F., Parsia, B., Bock, C., Fokoue, A., Haase, P., Hoekstra, R., Horrocks, I., Ruttenberg, A., and Sattler, U.: ‘Owl 2 web ontology language: Structural specification and functional-style syntax’, W3C recommendation, 2009, 27, (65), pp. 159

7 Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., and Rudolph, S.: ‘OWL 2 web ontology language primer’, W3C recommendation, 2009, 27, (1), pp. 123

8 Brachman, R.J., and Schmolze, J.G.: ‘An Overview of the KL - ONE Knowledge Representation System*’, Cognitive science, 1985, 9, (2), pp. 171-216

9 Baader, F.: ‘The description logic handbook: theory, implementation, and applications’ (Cambridge university press, 2003. 2003)

10 Organization, W.H.: ‘International classification of diseases (ICD)’, 2012

11 Singhal, A.: ‘Introducing the knowledge graph: things, not strings’, Official Google Blog, May, 2012

12 Antoniou, G., and Van Harmelen, F.: ‘Web ontology language: Owl’: ‘Handbook on ontologies’ (Springer, 2004), pp. 67-92

13 Noy, N., Rector, A., Hayes, P., and Welty, C.: ‘Defining n-ary relations on the semantic web’, W3C Working Group Note, 2006, 12, pp. 4

14 Ontology, P.: ‘Knowledge Acquisition System’, See http://protege. stanford. edu, 2007

15 Beebe, N.L., and Clark, J.G.: ‘A hierarchical, objectives-based framework for the digital investigations process’, Digital Investigation, 2005, 2, (2), pp. 147-167

16 http://protegewiki.stanford.edu/wiki/OntoGraf2014

17 facetmap.com/2014

18 Ellson, J., Gansner, E., Koutsofios, L., North, S.C., and Woodhull, G.: ‘Graphviz—open source graph drawing tools’, in Editor (Ed.)^(Eds.): ‘Book Graphviz—open source graph drawing tools’ (Springer, 2002, edn.), pp. 483-484

19 Fenz, S., and Ekelhart, A.: ‘Formalizing information security knowledge’, in Editor (Ed.)^(Eds.): ‘Book Formalizing information security knowledge’ (ACM, 2009, edn.), pp. 183-194

20 Osorno, M., Laurel, M., Millar, T., Team, E.R., and Rager, D.: ‘Coordinated Cybersecurity Incident Handling’, in Editor (Ed.)^(Eds.): ‘Book Coordinated Cybersecurity Incident Handling’ (2011, edn.), pp.

21 Magklaras, G., and Furnell, S.: ‘Insider threat prediction tool: Evaluating the probability of IT misuse’, Computers & Security, 2001, 21, (1), pp. 62-73

22 Wang, J.A., and Guo, M.: ‘OVM: an ontology for vulnerability management’, in Editor (Ed.)^(Eds.): ‘Book OVM: an ontology for vulnerability management’ (ACM, 2009, edn.), pp. 34

23 Chiang, T.J., Kouh, J.S., and Chang, R.-I.: ‘Ontology-based Risk Control for the Incident Management’, IJCSNS International Journal of Computer Science and Network Security, 2009, 9, (11), pp. 181-189

24 Division, O.o.C.a.C.N.C.S.: ‘Information Technology (IT) Security Essential Body of Knowledge (EBK): A Competency and Functional Framework for IT Security Workforce Development’, 2007

25 Lambo, T.: ‘ISO/IEC 27001: The future of infosec certification’, ISSA Journal, Information Systems Security Organization (http://www. issa. org), 2006

26 Ekelhart, A., Fenz, S., Klemen, M., and Weippl, E.: ‘Security ontologies: Improving quantitative risk analysis’, in Editor (Ed.)^(Eds.): ‘Book Security ontologies: Improving quantitative risk analysis’ (IEEE, 2007, edn.), pp. 156a-156a

27 Schlenoff, C., Gruninger, M., Tissot, F., Valois, J., Lubell, J., and Lee, J.: ‘The process specification language (PSL) overview and version 1.0 specification’ (Citeseer, 2000. 2000)

APPENDIX A: AN OVERVIEW OF THE ACTIVITIES CLASS IN THE ONTOLOGY

- Incident management (IM) leaders develop trusted relationships with external groups
- Both internal and external trainers provide training to IM personnel
- internal trainers provide awareness training to partners
- IM leaders develop governance artifacts
- IM leaders perform management functions on IM personnel
- IM leaders develop planning artifacts
- IM personnel provide vulnerability remediation to constituents
- IM personnel provide incident detection to constituents
- IM personnel provide incident communication to constituents
- IM personnel provide defense hardening to constituents
- IM personnel perform triage
- incident data collectors perform incident data collection
- IT personnel restore IT components
- IT personnel validate IT components
- IM personnel coordinate analyzing lessons learned
- IM incident handlers perform incident analysis
- IM personnel perform incident resolution
- IM leaders perform management functions
- IM personnel perform incident tracking

Appendix B: The Class Hierarchy of the IM Ontology

This appendix contains the class hierarchy in the Incident Management Ontology.

- **activities**: functions performed by a CSIRT
  - **developing-governance**: establishing the operational guidelines for an organization
    - **developing-plans**: establishing and maintaining the business and operational plans for an organization
    - **developing-policies**: establishing and maintaining the policies that guide the organizational activities
    - **developing-procedures**: establishing and maintaining implementations of organizational policies
    - **developing-processes**: establishing and maintaining organizational processes
    - **develop-data-collection-processes**: establishing logs and monitoring to provide insight into incidents
  - **developing-relationships**: identifying and communicating with essential business partners
    - **developing-external-relationships**: developing relationships with external parties
    - **developing-internal-relationships**: developing relationships with internal parties
  - **prepare-activities**: activities that are typically carried out during the prepare phase of the incident life cycle
  - **process-improvement**: activity whose goal is to improve the efficiency, reproducibility, reliability, or other quality attribute of business processes
    - **integrating-lessons-learned**: feeding the results of a postmortem review into the organization's problem-solving process
    - **postmortem-review**: an examination of an event to discover factors that affected the quality of the handling of the event
    - **measuring-performance**: collecting metrics that assess the quality of a process for process improvement purposes
  - **protect-activities**: activities that are typically carried out during the protect phase of the incident life cycle
  - **respond-activities**: activities that are typically carried out during the respond phase of the incident life cycle
  - **service-delivery**: the activity of providing a service to a constituent
    - **defense-hardening-service**: assisting with improving the security defenses of a constituent
      - **improve-defenses:** hardening defenses by improving the security controls in place
      - **remediation-service**: hardening defenses by removing known vulnerabilities and risks
      - **risk-assessment-service**: hardening defenses by identifying threats
      - **vulnerability-assessment-service:** hardening defenses by identifying vulnerabilities
    - **incident-response-service:** providing assistance in responding to and recovering from incidents
      - **evidence-collection-service:** gathering and maintaining information concerning an event
        - **diagnostic-data-collection-service:** incident-data-collection to support diagnosis and restoration activities
        - **forensics-data-collection-service:** incident-data-collection to support legal activities
      - **incident-analysis-services:** using collected data to uncover the causes and time-line of an event
        - **artifact-analysis-service:** incident analysis applied to artifacts
        - **incident-analysis-service:** general incident analysis
        - **root-cause-analysis-service:** incident analysis with the goal of determining the root cause of an event
        - **vulnerability-analysis-service:** incident analysis applied to the vulnerability that enabled an event
      - **incident-detection-service:** proactive steps to ensure events and incidents are discovered and reported as soon as possible
      - **incident-recovery-service:** reactive activities with the goal of restoring an affected system to the state before an event
        - **system-restoration-service:** restoring an affected system to the state before an event
        - **system-validation-service:** verifying that an affected system has been restored
      - **incident-reporting-service:** communicating information about an event or incident in accordance with an incident reporting policy
      - **monitoring-service:** maintaining an automated infrastructure to detect events and report incidents
    - **training-service:** a proactive service to ensure that stakeholders have the knowledge, skills, and abilities they need
      - **constituent-training-service:** training for constituents that helps them protect their infrastructure
      - **staff-training-service:** training for staff that helps them perform their jobs
        - **team-training-coordination:** ensuring adequate training for staff

  - **sustain-activities:** activities whose goal is to prevent the CSIRT's posture from declining over time
- **crosscuts:** constraints or principles that apply to activities
  - **incident-communication:** communicating information about the effects of a incident to staff and constituents
  - **incident-coordination:** ensuring that all IM stakeholders are with a shared plan
  - **incident-data-collection:** collection of data relevant to an incident
  - **incident-documentation:** documenting the results of incident-analysis
  - **incident-problem-solving:** using generic or specialized methods in an orderly manner to find solutions to problems
  - **incident-resolution:** an action taken to repair the root cause of an incident or to implement a workaround
  - **incident-tracking:** managing and maintaining a database of information on incidents and constituents
- **incident-components:** the various elements that constitute the conceptual model of an event
  - **artifacts:** any entities left behind after an incident takes place; for example, malicious code or logfiles
  - **events:** any occurrences that may have negative security consequences
  - **incidents:** events that have been confirmed to have negative security consequences
  - **root-causes:** the earliest occurrence in the causal chain leading to an incident
  - **vulnerabilities:** the weaknesses in the system that were exploited by an incident
- **IT-components:** the various elements that constitute the conceptual model of an IT system
  - **information-system:** collection of technical and human resources that provide storage, computing, and distribution for enterprise information
  - **network:** collection of host computers together with the sub-network or inter-network through which they can exchange data
  - **security-tools**: hardware and software that improves the security of the information-system in which they are installed
    - **incident-detection-tools:** security-tools that perform incident-detection
      - **av-systems:** incident-detection-tools that work by analyzing virus signatures
      - **ids-systems:** incident-detection-tools that work by analyzing activity on the network
    - **network-monitors:** security-tools that work by observing network activity
- **knowledge-assets:** the various types of documents that constitute the intellectual capital of the organization
  - **governance-artifacts:** documents that are used in the process of governing
    - **policies:** abstract documents that express decisions made by management about the running of the organization
    - **procedures:** concrete documents that implement policies
    - **processes:** workflows that implement policies and procedures
  - **incident-reports:** documents that inform the CSIRT about events and incidents
  - **incident-tracking-documents**: case management documents that trace the progress of an event through the incident-handling process
    - **incident-assignments:** tagging of incidents with the names of IM-personnel responsible for handling them
    - **incident-categorization:** tagging of incidents with the classification into which they fall
  - **information:** general documents that do not fall in any other category
  - **lessons-learned:** documents that capture the results of analyzing-lessons-learned
  - **other-knowledge-assets:** any information not included in other categories
  - **planning-artifacts:** abstract documents that prepare IM-personnel for incident response
    - **incident-response-plans:** planning-artifacts that reflect decisions made about incident-response within the organization
    - **incident-response-strategies:** technical documents that guide IM-personnel in responding to incidents
  - **training-materials:** documents that are used to provide training
- **life-cycle-phase:** the temporal periods into which incident response is divided
  - **prepare-phase:** educating personnel and providing them with the tools needed to perform their jobs
  - **protect-phase:** applying controls and otherwise hardening the infrastructure to resist attack
  - **respond-phase:** detecting, analyzing, and recovering from incidents
  - **sustain-phase:** ensuring that the capability of the CSIRT does not degrade over time
- **organizational-groups:** stakeholders in the incident management process
  - **external-groups:** stakeholders not within the administrative boundaries of the organization
    - **external-csirts:** incident management teams outside the boundaries of the organization
    - **external-trainers:** educational personnel outside the organization
    - **law-enforcement-agencies:** external groups performing law enforcement functions
    - **other-external-groups:** any other external group
  - **partners:** groups or sets of individuals with close relationships to the organization

        - **constituents:** the groups or sets of individuals for whom incident management is being performed
    - **staff:** stakeholders contained within administrative boundaries of the organization
        - **IM-personnel:** groups or sets of individuals tasked with performing incident management
            - **IM-incident-handlers:** individuals responsible for responding to and recovering from incidents
                - **IM-forensics-analyst:** an IM-incident-analyst specializing in analysis for legal purposes
                - **IM-incident-analyst:** an IM-incident-handler specializing in the analysis of incident-components
                - **IM-malware-analyst:** an IM-incident-analyst specializing in reverse engineering
            - **IM-leaders:** individuals responsible for leading the incident management personnel
        - **incident-data-collectors:** individuals responsible for collecting data about incidents
            - **diagnostic-data-collectors:** incident-data-collectors that collect data for diagnostic purposes
            - **forensic-data-collectors**: incident-data-collectors that collect data for forensic purposes
        - **internal-trainers:** educational personnel within the organization
        - **IT-personnel:** members of the it staff that carry out security functions such as infrastructure hardening
        - **management:** individuals responsible for governing
            - **line-management:** managers at the low end of the chain of command
            - **mid-level-management:** managers in the middle of the chain of command
            - **senior-management:** managers at the high end of the chain of command
        - **risk-assessors:** individuals responsible for assessing risks to the organization
        - **vulnerability-assessors:** individuals responsible for identifying vulnerabilities in the organization's infrastructure
- **quality-standards:** normative requirements for ensuring the high quality of the CSIRT's activities
    - **document-management-standards:** standards that constrain the way information is handled within the organization
        - **appropriately-dissemination-standards:** standards that govern the provision of information to the appropriate audiences
        - **confidentiality-preserving-standards:** standards that govern how information is withheld from inappropriate audiences
    - **forensic-standards:** quality standards that ensure the admissibility of the analysis in a court of law
        - **preserving-chain-of-custody**: documenting that there has been no opportunity for forensic evidence to be tampered with
    - **other-quality-standards:** quality standards not included in other categories
- **relationships:** connections between individuals or groups of individuals
    - **trusted-relationship:** relationships among entities that are willing to share confidential data
    - **untrusted-relationship:** relationships among entities that are willing to share confidential data
- **team-resources:** anything needed for the CSIRT activities or the operations of IM-personnel
    - **funding:** financial resources necessary for the operations of IM-personnel
    - **IT-infrastructure:** information security assets necessary for the operations of IM-personnel
    - **staffing**: human resources necessary to ensure the operations of IM-personnel

An Ontology for Medical Treatment Consent

Bo Yu, Duminda Wijesekera and Paulo Costa
Department of Computer Science
George Mason University,
Fairfax VA, USA
{byu3, dwijesek, pcosta}@gmu.edu

Abstract— **Active duty military personnel, their families and veterans seek medical services from the Military Health Service, which partners with private care, or the Veterans Administration, respectively. Indeed, medical services for active duty personnel, who need medical services on deployment, is a readiness issue. Laws that govern the practice of medicine, licensing to practice medicine and the permission to treat a patient is based on local laws (state level) that are specific to medical sub-specialties. That provides a daunting challenge to patients who move frequently, such as active duty military and their families. As most medical providers are transforming their record keeping to Electronic Medical Record (EMR) system, it is desirable to obtain, verify and act according to the legally enforced medical consent using EMRs. We present an Ontology-based framework and a prototype system that provide end-to-end services using an open source EMR system. Providing an electronically verifiable, but compliant with locally mandated laws in one universal system can be beneficial to VA and other DoD EMR systems.**

Keywords—informed medical consent; medical consent law; workflow management system; ontology

I. INTRODUCTION

Failure to obtain informed consent is listed as a top ten reason for medical malpractice claims [1]. The improvement in flexibility, automation and enforcement for electronic patient informed consent management are especially beneficial to patients who relocate, such as active duty military and their families. This mobility entails their medical treatment be subject to local regulations. Given that EMRs services can be centralized, cloud based or being offered remotely, having a consent management system that can provide a diverse collection of consents for every treatment would benefit EMR services generally, and especially the Military Health Service. Although some VA hospitals have implemented electronic consent process, iMedConsent [2], they do not provide enforcement mechanism and is considered mostly educational for the patients. The system we prototype can accommodate (i.e. obtain and enforce though out long chains of treatment processes), can be deployed from one location but cover multiple regions (such as states, countries) and be helpful for the military, military dependants and as well as for other mobile populace.

Informed patient consent – either express or derived -- expresses the patient's wishes, and consists of an agreement between the care providers and patient, including choice between potential treatment regimes or terminating treatment. Part of the process of obtaining consent involves the caregiver providing a risk/benefit analysis and explaining alternative treatments in a way that the patient understands, and accurately communicates the care provider's understanding in an unbiased way [3].

State law specifies acceptable explanation. Further, consent laws obligate the caregiver to attest that the patient and/or the guardian have the capacity (including physical/mental capacity and maturity) to provide consent. Over the years, federal, state, and local governments and healthcare organizations have developed laws, regulations, and standards for obtaining and memorializing informed consent. However, consent laws and regulations are complex and sometimes ambiguous, and change often. Therefore EMR must take these changes as they are mandated. We postulate that having a consent service that is aware of the semantics of informed medical consent can satisfy the evolving and diverse nature of mandated informed treatment consents.

As a substantiation of our postulate, we provide a semantic web driven, medical workflow aware [4] control system to obtain and enforce treatment consent. The medical personnel that use our system do not see a difference between the existing EMR system and our prototype. Some highlights of our system are: A refined Workflow-based EMRs that allow the medical staff to obtain consents dynamically--i.e., if required by a procedure in a treatment workflow; and evaluating these consents automatically as a care team goes from one step to another in the treatment workflow [5]. Furthermore, our combined workflow based consent management engine ensures that treatment workflow move forward only if consents have been granted (including break-the-glass kind of emergency treatments). This enhancement improves current practice of patient informed consent management.

Following this Introduction, Section 2 describes related work; Section 3 explores ontology-based reasoning to derive the informed treatment consent; Section 4 shows architecture of our consent-based workflow control in a Workflow-based EMR system; and finally, Section 5 contains concluding comments.

II. RELATED WORKS

A. *Informed Consent in Current EMRs*

The American Medical Association considers the term *informed consent*, first used by a California appeals court in 1957 [6], "an ethical obligation of the practice of medicine and a legal requirement per statute and case law in all 50 States" [7] y. Medical informed consent falls mainly into two categories: consent for medical information disclosure; and consent for medical treatments. Herein we mainly address the latter, with a focus on informed consent for procedure-oriented treatment regimes.

In the past decade, consent management has received considerable attention from researchers and healthcare organizations who proposed different ways to improve electronic consent management system. For example, "e-Consent: The Design and Implementation of Consumer Consent Mechanisms in an Electronic Environment" [8] provided guidelines on how to design an e-consent system. Another relevant work is by Ruan C. & Yeo S.S. [9], who used the UML Model to design an e-consent system. They first identify various parts necessary to specify the e-Consent rules about patient record protection, and then used UML to model the properties required by an e-consent system and to make the associated patient record protection rules explicit and verifiable. However, that work was theoretical; they neither designed nor implemented a system that works with EMR systems.

Rusello G. et al. proposed creating consent-based workflows for healthcare management [10] where patients can control disclosure of their medical data for inter-institutional consults. This work does not address workflows for procedure-oriented treatment regimes, treating consent contents as black boxes. Others have proposed e-consent management to be integrated with EMR or EHR systems [11-14]. Win et al. in their paper "Implementing patients consent in electronic health record systems" [15] expressed patient consent using an interface-based approach. However, those e-consent approaches focus mainly on sharing medical data, privacy, and security aspects [16-18], but not the complicated nature of treatments.

Many healthcare organizations attempted to have electronic consent management in their EMRs. Veterans Administration Medical Centers use iMedConsent™ [2] that supports electronic access, completion, signing, and storage of informed consent forms and advance directives. iMedConsent has two parts: software application and clinical content library. It generates consents on each procedure without workflows. Nonetheless, the system neither dynamically gains informed consents at the point of providing treatments nor enforces consents on medical procedures.

B. *Ontologies in the Healthcare Domain*

Ontologies have been used to represent actionable knowledge in biomedicine [19–23], decision support [24], information integration, etc. Some examples are: BioPAX, an ontology for the exchange and interoperability of biological pathway (cellular processes) data [25]; CCO and GexKB, Application Ontologies (APO) that integrate diverse types of knowledge with the Cell Cycle Ontology (CCO) and the Gene Expression Knowledge Base (GexKB) [26]; Disease Ontology, designed to facilitate the mapping of diseases and associated conditions to particular medical codes [27]; Linkbase, a formal representation of the biomedical domain, founded upon Basic Formal Ontology [28]; NCBO Bioportal, biological and biomedical ontologies and associated tools to search, browse and visualize [29]; NIFSTD Ontologies from the Neuroscience Information Framework: a modular set of ontologies for the neuroscience domain [30]; SNOMED CT (Systematized Nomenclature of Medicine -- Clinical Terms) [31]; OBO Foundry, a suite of interoperable reference ontologies in biology and biomedicine [32]; OBO-Edit, an ontology browser for most of the Open Biological and Biomedical Ontologies [33]; PRO, the Protein Ontology of the Protein Information Resource from Georgetown University [34], and so on. Yet, no works have efficiently leveraged a technique for informed treatment consent in EMRs. In this paper, we provide a methodology to address this gap.

III. USING ONTOLOGY-BASED REASONING TO DERIEVE INFORMED TREATMENT CONSENTS

A. *Entities of Medical Treatment Consent Ontology*

To create our ontology for medical treatment consents, we studied several medical treatments in actual medical facilities, obtained their consent forms and studied state law governing medical consents. We combined information obtained from interviews with the various paper-based documents used to record events and data that are associated with the workflows. We found there are common entities used in the informed treatment consents, such as patients (may or may not be an *Informed consent giver*), treatments (usually, consisting of several treatment procedures – so called tasks in the treatment workflow specifications), treatment performance locations (some treatments may be not be permitted in some states) and informed consents (where some procedures within a treatment regime may not require consent). Based on our observations, we created the following classes, attributes and rules on the ontologies.

B. *Classes, Propertities Created in Ontology*

➢ Classes

1. Patient: (one requiring medical assistance) with attributes such as age, name and active status used to evaluate maturity.

2. Treatment: Methods used to manage ameliorate, or prevent a disease, disorder, or injury. Each Treatment has a name (such as eye surgery, dialysis etc.).

3. Procedures: generally, every treatment consisted of a set of predefined procedures. Each procedure has a procedure's name.

4. Consent: legal documents expressing the willingness for the patient to be subjected to treatments and encompassing procedures (referred to as TreatmentConsent) or providing the authority share medical information (SharingConsent).

5. TreatmentConsent: A subclass of Consent, modeling the agreement to receive treatment. Its nature is determined by state law, federal law or medical sub-discipline. Thus, the attributes are the state, treatment name, treatment type. An example, *anesthesiaConsent*

Identify applicable sponsor/s here. If no sponsors, delete this text box (*sponsors*).

1) MandatoryConsent: a sub-class of TreatmentConsent with attributes active (or passive). An example is *anesthesiaConsent* for Suegery.

2) OptionalConsent: is a sub-class of TreatmentConsent, but its omission does not affect performing the procedures. An example is anesthesia consent for giving birth. Most states do not mandate this consent.

6. AdultPatient: is the patient's maturity status. Competent adult patients may give their own treatment consents.

7. MinorPatient: is a patient's maturity status. Without exception, such as during an emergency, minor patients cannot provide treatment consent.

8. PerformInState: is a State in which the treatment is to be performed. They associate with Treatment.

➢ Properties (express the relationship of two classes) in Ontology

TABLE 1 PROPERTIES TABLE

Property Name	Domain	Range
asksMandatoryConsentByPatient	Patient class	MandatoryConsent
asksOptionalConsentByPatient	Patient class	OptionalConsent class
has	Treatment class	Procedures class
isPatient	AdultPatient class or MinorPatient class	Patient class
isState	PerformInState class	State class
needsMandatoryConsent	Procedures class	MandatoryConsent class
needsOptionalConsent	Procedures class	OptionalConsent class
performedIn	Treatment class	State class
requiresMandatoryConsent	Procedures class	Consent class
requiresOptionalConsent	Procedures class	Consent class

Table 1 shown relationship between two classes. Properties may have a domain and a range specified. For example, row1 in above table indicates:

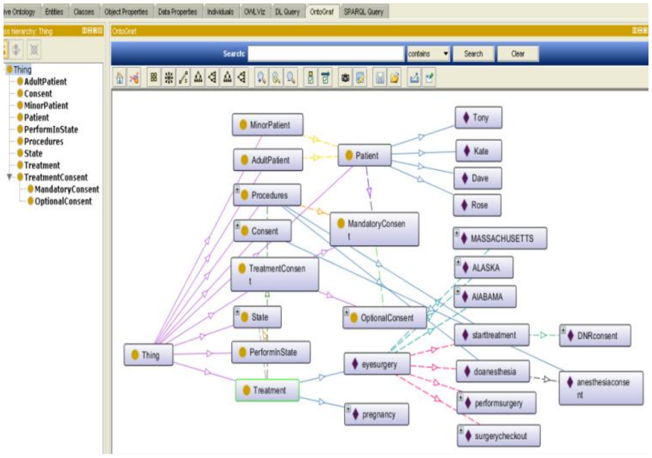
asksMandatoryConsentByPatient: it links individuals belonging to the class Patient to individuals belonging to the class MandatoryConsent.

A view of the entities of treatment consent ontology developed in Protégé 4.3. shown in Fig.1.

C. Rules for Enforcing Informed Treatment Consent

We now show how to use the ontological syntax and create rules that specify treatment consent. As stated, these rules formalize contents taken from the many natural language documents consisting of state laws and sub-disciplines regulations that govern specific institutional practices [35]. These rules specify in the consent components:

Fig. 1. Entities of treatment consent ontology



Rule (1) *Information Disclosure Standard:* Obligates the care provider to disclose and discuss information relevant to the proposed treatment, their risks and benefits and the available alternatives with their risks and benefits [36]. These come in two main standards: The normal person's standard and the professional standard. 25 states mandate the use of the patient standard, while 23 have mandated the professional standard. The laws in the remaining two states, Colorado and Georgia, are not easily classifiable as one or the other [37]. Nonetheless, the scope of required information to be disclosed is still being debated. Two states, Minnesota and New Mexico, require the care provider to explain using both these standards.

Rule (2) *Decisional Capability:* Evaluation of patient's competence to understand the information and providing rational and voluntary decisions about the healthcare treatment. In [38], authors described four psycho-legal standards, communicating a choice, factual understanding, appreciation of the situation, and rational manipulation of information, all used to evaluate a patient's competence in giving consent. However, to date this lacks a widely accepted standard. Hence, we do not codify this aspect.

Rule (3) *Competency:* Validation of patient's maturity to grant informed consent. For the informed treatment consents, an essential component of the conception of autonomy is allowing competent adult persons and emancipated children to make their own health care decisions. Our examinations have led to categorizing the consents as follows:

1. *Informed consent giver (governed by* Rule (3) *- competence):* the person with the legal right to make health care decisions, such as parents or legal guardians of minors, healthcare proxies, healthcare providers or third parties.

2. *Treatment information (governed by* Rule (1) *- information or disclosure):* at a minimum, includes treatment name, procedures for this treatment, treatment preformed location.

3. *Patient's decision of the treatment (governed by* Rule (2) *- decisional capability):* includes the decision

(deny or accept) by providing all required conditions such as patient's and other attributes such as signatures, date, etc.

Consequently, formalization of informed consent should base its consents on all the above-mentioned attributes. Assuming that consent rules and patient information is available in an EMR, we show how to generate the consent decisions. Auto-generation of the appropriate forms to be signed by the consent giver will be described elsewhere.

The following example shows the complicated nature of decisions made by our consent service. Most states set the age at 18 years, but Alabama allows health care consent to be made by minors 19 years of age and older [39]. So, can an 18 year-old resident of Virginia requiring dialysis treatment during a visit to Alabama give consent for the treatment? Answering this question will determine the adult status of the VA resident, but that too depends on the treatment sought as described below.

– Depending on the treatment type, the age of the minors who may consent may differ.

Example: In CA, for General Medical Treatments, Cal. Fam. Code § 6500, states a minor 18 years of age or older may give his/her own treatment consent. However, for Pregnancy (not include sterilization and abortion), CAL. FAM. CODE § 6925 (2012) states that *a minor may consent to medical care related to the prevention or treatment of pregnancy, but this law does not authorize a minor: (1) To be sterilized without the consent of the minor's parent or guardian. (2) To receive an abortion without the consent of a parent or guardian other than* as provided in Section 123450 of the Health and Safety Code.

– Even if the patients are minors, for certain treatment with some minor active status such minors are allowed to give their own treatment consent.

Example: (1) Cal. Fam. Code § 7050 provides that *an emancipated minor may consent for medical, dental, or psychiatric care, without parental consent, knowledge, or liability*; (2) Cal. Fam. Code § 6922 provides that *a minor, 15 years of age or older, is living separate and apart from the minor's parents or guardian, whether with or without the consent of a parent or guardian and regardless of the duration of the separate residence; and the minor is managing the minor's own financial affairs, regardless of the source of the minor's income can give consent for medical treatments*.

– Some consent rules are not found in specific provision explicitly, but can be retrieved from combining laws.

Example: Cal. Fam. Code § 7002 provides *a minor who has married is emancipated*; according to another rule (Cal. Fam. Code § 7050 provides that *an emancipated minor may consent for medical, dental, or psychiatric care, without parental consent, knowledge, or liability*). The combination implies *a married minor may consent for medical, dental, or psychiatric care, without parental consent, knowledge, or liability*.

We create patient maturity evaluation rules for each state based on its consent laws. Table 2 shows a part of the summary of 50 states' patient maturity evaluation rules.

TABLE 2 PATIENT MATURITY EVALUATION RULES (*50 STATES*)

State	State Abbreviation	General Medical Treatment	Pregnancy
ALABAMA	AL	19 years of age or older (Ala. Code § 26 1 1)	
		1. Minor age equal or greater than 18, less than 19, and minor has an emancipation order (Ala. Code §§ 26-13-1 and 26-13-5); 2. Minor age 14 or old, has graduated from high school (Ala. Code § 22-8-4); 3. Minor is married (Ala. Code § 22-8-4; Ala. Code § 22-8-5); 4. Minor having been married and divorced (Ala. Code § 22-8-4; Ala. Code § 22-8-5); 5. Minor is pregnant (Ala. Code § 22-8-4); 6. Minor has child(ren) (Ala. Code § 22-8-5);	1. Any minor (Ala. Code § 22-8-6);
• • • • • •			
CALIFORNIA	CA	18 years of age or older (Cal. Fam. Code § 6500)	
		1. Minor is an emancipation minor (Cal. Fam. Code § 7050); 2. Minor is 15 years of age or older, who is living separate and apart from the minor's parents or guardian and managing the minor's own financial affairs (Cal. Fam. Code § 6922); 3. Married Minor is an emancipation minor (Cal. Fam. Code § 7002); 4. Minor is 16 years of age or older, who serves in the armed forces of the United States or has court order is an emanicpated minor (CAL. FAM. CODE § 6950 (2012));	1. An unemancipated minor (Cal. Fam. Code § 6925);
• • • • • •			
WYOMING	WY	18 years of age or older (Wyo. Stat. Ann. § 14 1 101(a))	
		1. Minor is or was legally married – minor is married, widow (Wyo. Stat. Ann. § 14-1-101(b)); 2. Minor is or was legally married – minor is divorced (Wyo. Stat. Ann. § 14-1-101(b)); 3. Minor who is in active military service of the United States may consent for healthcare treatment (Wyo. Stat. Ann. § 14-1-101(b)); 4. Minor who is living apart from his parents or guardian and managing his/her own affairs may consent for healthcare treatment(Wyo. Stat. Ann. § 14-1-101(b)); 5. Minor is an emancipated minor (Wyo. Stat. Ann. § 14-1-101(b));	1. No explicit law

D. Deriving Informed Treatment Consents

We use the patient maturity rules of California (CA) as an example to explain Semantic Web Rule Language (SWRL) rules:

- For General Treatment (we consider eye surgery belongs to general treatment)
  1. Minor is an emancipation minor may consent for medical, dental, or psychiatric care, without parental consent, knowledge, or liability. (Cal. Fam. Code § 7050);
  2. Minor is 15 years of age or older, who is living separate and apart from the minor's parents or guardian and managing the minor's own financial affairs (Cal. Fam. Code § 6922)is an emancipation minor;
  3. Married Minor is an emancipation minor (Cal. Fam. Code § 7002);
  4. Minor is 16 years of age or older, who serve in the armed forces of the United States or has court order(Cal. Fam. Code § 6950);

- For Pregnancy Treatment (exclude to be Sterilization and to receive Abortion)
  1. An un-emancipated minor may consent for medical care related to the prevention or treatment of pregnancy (Cal. Fam. Code § 6925);

Let S be a SWRL knowledge base, where {*t, p, s*} is a set of OWL class names. In here, {*t, p, s*} refers to {Treatment, Patient, and State} coordinately. *performedIn* is an OWL property name to show the relationship between Treatment and State, and {*"eyesurgery", "CA", age, fi, ls, m, iem, iaf, hco, tpi*} is a set of OWL constants and SWRL variables. In here, age refers to patient's age; *fi* refers to patient's financial status; *ls* refers to patient's resident status; *m* refers to patient's marital status; *iem* refers to patient maturity level; *iaf* refers to patient's career status; *hco* refers to a legal issue related to patient, *tpi* refers to patient seeking treatment which is an attribute of Patient. Some SWRL rules have the form:

Example 1: (CA consent Laws for General Medical Treatment: rule2 shown in Table 2)

(1) patientRequiresTreatment(?p, "eyesurgery"),
hasAge(?p, ?age),
patientFinancialIndependent(?p, ?fi),
patientLivesSeparately(?p, ?ls),

(2) hasTreatmentName(?t, "eyesurgery"),
patientTreatmentPerformedIn(?p, ?tpi),
hasStateName(?s, ?tpi), performedIn(?t, ?s),
containsIgnoreCase("AL || AK || CA || MA",
?tpi),

(3) containsIgnoreCase("T", ?fi),
containsIgnoreCase("T", ?ls),

(4) lessThan(?age, 16),
greaterThanOrEqual(?age, 15)

(5) -> AdultPatient(?p)

Example 2: (CA consent Laws for General Medical Treatment: rule1 ~ rule4 shown in Table1)

(1) patientRequiresTreatment(?p, "eyesurgery"),
hasAge(?p, ?age),
patientFinancialIndependent(?p, ?fi),
patientLivesSeparately(?p, ?ls),
patientMarried(?p, ?m),
patientIsEmancipatedMinor(?p, ?iem),
patientIsArmedForce(?p, ?iaf),
patientHasCourtOrder(?p, ?hco),
patientIsEmancipatedMinor(?p, ?iem),

(2) hasTreatmentName(?t, "eyesurgery"),
patientTreatmentPerformedIn(?p, ?tpi),
hasStateName(?s, ?tpi), performedIn(?t, ?s),
containsIgnoreCase("AL || AK || CA || MA",
?tpi),

(3) stringConcat(?v, ?fi, ?ls),
containsIgnoreCase("FF-FT-TF", ?v),
containsIgnoreCase(?iem, "F"),
containsIgnoreCase("F", ?m),
containsIgnoreCase("T-F", ?iaf),
containsIgnoreCase("T-F", ?hco),

(4) lessThan(?age, 16),
greaterThanOrEqual(?age, 15)

(5) -> MinorPatient(?p)

In Part (1) we defined a set of OWL constants and SWRL variables of a specific patient; and the information we can retrieve from EMRs. Part (2) checked whether the treatment that patients seek may be performed in the state where patient does the treatment; and which treatment can be performed in which states is known information. Part (3) established rules. Part (4) provided constrains. Part (5) implied the consequent ((5)) from the antecedent ((1) ~ (4)).

Table 3 shows the part of summary of the syntax of consent laws of patient's maturity in 50 states.

TABLE 3 THE SYNTAX OF CONSENT RULES OF PATIENT MATURITY IN 50 STATES

State	State Abbreviation	SWRL rule	
		General Medical Treatment	*Pregnancy*

ALABAMA	AL	hasTreatmentName(?t, "eyesurgery"), patientRequiresTreatment(?p, "eyesurgery"), hasAge(?p, ?age), patientTreatmentPerformedIn(?p, ?tpi), hasStateName(?s, ?tpi), performedIn(?t, ?s), containsIgnoreCase("AL", ?tpi), greaterThanOrEqual(?age, 19) -> AdultPatient(?p)
		hasTreatmentName(?t, "eyesurgery"), patientRequiresTreatment(?p, "eyesurgery"), hasAge(?p, ?age), patientDivorced(?p, ?d), patientIsPregnant(?p, ?ip), patientMarried(?p, ?m), patientHasChild(?p, ?hc), stringConcat(?v, ?m, ?d, ?ip, ?hc), containsIgnoreCase(?v, "T"), patientTreatmentPerformedIn(?p, ?tpi), hasStateName(?s, ?tpi), performedIn(?t, ?s), containsIgnoreCase("AL", ?tpi), lessThan(?age, 19) -> AdultPatient(?p)
		⋮
			hasTreatmentName(?t, "pregnancy"), patientRequiresTreatment(?p, "pregnancy"), patientTreatmentPerformedIn(?p, ?tpi), hasStateName(?s, ?tpi), performedIn(?t, ?s), containsIgnoreCase("AL", ?tpi) -> AdultPatient(?p)

••••••			

CALIFORNIA	CA	hasTreatmentName(?t, "eyesurgery"), patientRequiresTreatment(?p, "eyesurgery"), hasAge(?p, ?age), patientTreatmentPerformedIn(?p, ?tpi), hasStateName(?s, ?tpi), performedIn(?t, ?s), containsIgnoreCase("CA", ?tpi), greaterThanOrEqual(?age, 18) -> AdultPatient(?p)
		hasTreatmentName(?t, "eyesurgery"), patientRequiresTreatment(?p, "eyesurgery"), hasAge(?p, ?age), patientFinancialIndependent(?p, ?fi), patientLivesSeparately(?p, ?ls), patientMarried(?p, ?m), patientIsEmancipatedMinor(?p, ?iem), patientIsArmedForce(?p, ?iaf), patientHasCourtOrder(?p, ?hco), containsIgnoreCase("F", ?hco), patientIsEmancipatedMinor(?p, ?iem), stringConcat(?u, ?fi, ?ls), containsIgnoreCase("FF-FT-TF", ?u), stringConcat(?v, ?iaf, ?hco), containsIgnoreCase("FF-FT-TF-TT", ?v), containsIgnoreCase(?iem, "F"), containsIgnoreCase(?m, "F"), patientTreatmentPerformedIn(?p, ?tpi), hasStateName(?s, ?tpi), performedIn(?t, ?s), containsIgnoreCase("CA", ?tpi), lessThan(?age, 16), greaterThanOrEqual(?age, 15) -> MinorPatient(?p)
			hasTreatmentName(?t, "pregnancy"), patientRequiresTreatment(?p, "pregnancy"), patientTreatmentPerformedIn(?p, ?tpi), hasStateName(?s, ?tpi), performedIn(?t, ?s), containsIgnoreCase("CA", ?tpi) -> AdultPatient(?p)

		⋮	
••••••			
WYOMING	WY	hasTreatmentName(?t, "eyesurgery"), patientRequiresTreatment(?p, "eyesurgery"), hasAge(?p, ?age), patientMarried(?p, ?m), patientDivorced(?p, ?d), patientIsArmedForce(?p, ?iaf), patientIsEmancipatedMinor(?p, ?iem), stringConcat(?v, ?m, ?d, ?iaf, ?iem), containsIgnoreCase(?v,"T"), patientTreatmentPerformedIn(?p, ?tpi), hasStateName(?s, ?tpi), performedIn(?t, ?s), containsIgnoreCase("WY", ?tpi), lessThan(?age, 18) -> AdultPatient(?p) ⋮	1. No explicit law

E. Evaluation

Here, we show consequences of our rule base that comply with state consent laws and sub-disciplines regulations. The scenario of a use case is a 15 year-old patient named Kate seeking eye surgery in California. She is not married nor has she done an emancipated minor evaluation. She also does not have a court order of giving medical consent nor is serving in the U.S. Armed Forces. However, she does not live with her parents and manages her own financial affairs. In this situation, what kind of informed consents should be obtained by her care providers? May she provide these consents herself? We derive that Kate is an adult patient according to CA consent laws of patient's maturity. Therefore, she is able to consent by herself, even if her age is under CA's required maturity age.

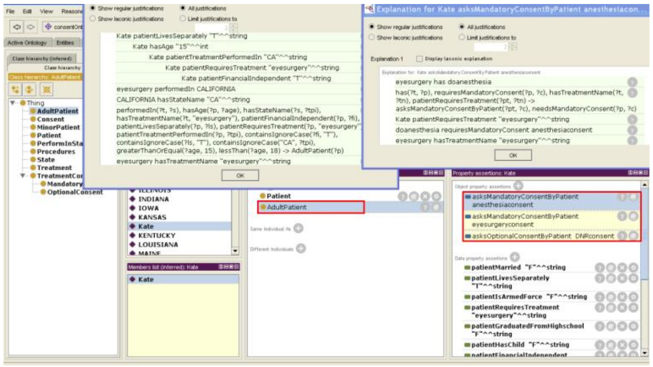
We now show how Pellet generates data properties of an individual of class Patient, here Kate, and object properties of this individual, reasoned with rules to infer the head of rule (see example 1).

Using Pellet, the informed treatment consents retrieved easily and appropriately. The outcome of the proof of patient maturity and explanation is shown in Fig. 2. In this illustration, the left red box exposed that the outcome matches our presuming result. For more details of how Pellet reasons, see the following explanation provided by Protégé.

Explanation for: Kate Type AdultPatient

1. Kate has Age "15"^^ int
2. Kate patientRequiresTreatment "eyesurgery"^^string
3. Kate patientTreatmentPerformedIn "CA"^^string
4. Kate patientFinancialIndependent "T"^^string
5. Kate patientLivesSeparately "T"^^ string
6. eyesurgery hasTreatmentName "eyesurgery"^^string
7. CALIFORNIA hasStateName "CA"^^ string
8. eyesurger performedIn CALIFORNIA
9. performedIn(?t, ?s), hasAge(?p, ?age), hasStateName(?s, ?tpi), hasTreatmentName(?t, "eyesurgery"), patientFinancialIndependent(?p, ?fi), patientLivesSeparately(?p, ?ls), patientRequiresTreatment(?p, "eyesurgery"), patientTreatmentPerformedIn(?p, ?tpi), containsIgnoreCase(?fi, "T"), containsIgnoreCase(?ls, "T"), containsIgnoreCase("CA", ?tpi), greaterThanOrEqual(?age, 15), lessThan(?age, 18)

Fig. 2. Outcome of the proof of patient maturity using Pellet reasoner



● Show regular justifications ● All justifications
○ Show laconic justifications ○ Limit justifications to

Kate patientLivesSeparately "T"^^string
Kate hasAge "15"^^int
Kate patientTreatmentPerformedIn "CA"^^string
Kate patientRequiresTreatment "eyesurgery"^^string
Kate patientFinancialIndependent "T"^^string
eyesurgery performedIn CALIFORNIA
CALIFORNIA hasStateName "CA"^^string
performedIn(?t, ?s), hasAge(?p, ?age), hasStateName(?s, ?tpi), hasTreatmentName(?t, "eyesurgery"), patientFinancialIndependent(?p, ?fi), patientLivesSeparately(?p, ?ls), patientRequiresTreatment(?p, "eyesurgery"), patientTreatmentPerformedIn(?p, ?tpi), containsIgnoreCase(?fi, "T"), containsIgnoreCase(?ls, "T"), containsIgnoreCase("CA", ?tpi), greaterThanOrEqual(?age, 15), lessThan(?age, 18) -> AdultPatient(?p)
eyesurgery hasTreatmentName "eyesurgery"^^string

Explanation for Kate asksMandatoryConsentByPatient anesthesiacon...

● Show regular justifications ● All justifications
○ Show laconic justifications ○ Limit justifications to

Explanation 1 □ Display laconic explanation

Explanation for: Kate asksMandatoryConsentByPatient anesthesiaconsent
eyesurgery has doanesthesia
has(?t, ?p), requiresMandatoryConsent(?p, ?c), hasTreatmentName(?t, ?tn), patientRequiresTreatment(?pt, ?tn) -> asksMandatoryConsentByPatient(?pt, ?c), needsMandatoryConsent(?p, ?c)
Kate patientRequiresTreatment "eyesurgery"^^string
doanesthesia requiresMandatoryConsent anesthesiaconsent
eyesurgery hasTreatmentName "eyesurgery"^^string

Property assertions: Kate

Object property assertions
asksMandatoryConsentByPatient anesthesiaconsent
asksMandatoryConsentByPatient eyesurgeryconsent
asksOptionalConsentByPatient DNRconsent

Data property assertions
patientMarried "F"^^string
patientLivesSeparately "T"^^string
patientIsArmedForce "F"^^string
patientRequiresTreatment "eyesurgery"^^string
patientGraduatedFromHighschool "F"^^string
patientHasChild "F"^^string
patientFinancialIndependent

In sub-section D above, we reviewed these rules, see Example 1. The input facts of individual patient, Kate, are shown in line 1 ~ line 6 from Kate's data prosperities; line 9 is the rule that used by Pellet to infer the new fact, in other words Kate belongs to adult patient base on her active status based on this particular rule.

Our goals are proposing a novel approach, named Workflow-based EMRs with a consent management component to allow gaining informed treatment consents required by a procedure in a treatment workflow dynamically, and reasoning these consents automatically by using ontologies to ensure those consents comply with consent laws and regulations.

IV. *WORKFLOW-BASED EMRS WITH CONSENT MANAGEMENT*

To achieve our goals, we proposed a prototype, shown in Fig. 1. We develop a consent management component incepted Workflow-based EMRs which refers back to our previous works.

The existing EMRs lack a mechanism for dynamically obtaining appropriate informed treatment consents and lack a standard way for specifying, updating and checking compliance with governmental consent laws and sub-discipline regulations. Our goal here is to build a novel EMRs by adopting a variety of technologies to address this gap.

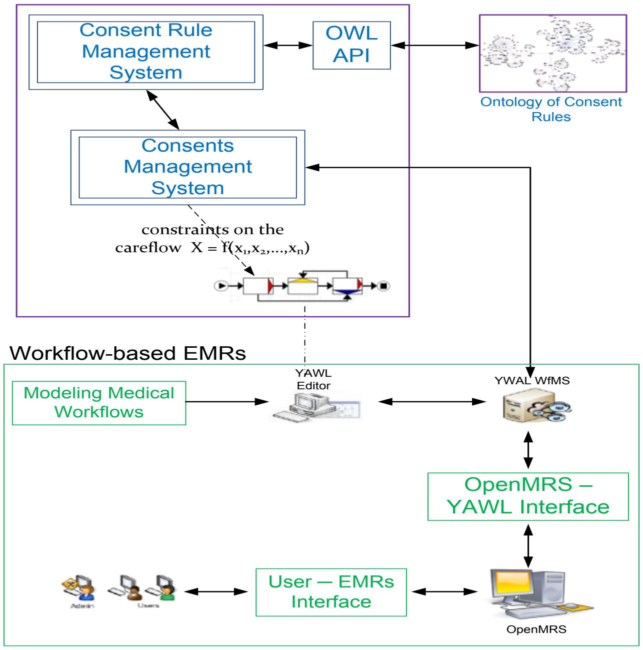
We developed a prototype consent management system on a Workflow-based EMR system. In our system, consents are issued electronically using the EMR interface and enforced using the workflow runtime. Furthermore, those consents can be used to control corresponding medical procedures dynamically. In addition, we use ontology-based knowledge representation and reasoning mechanisms to obtain required informed consents based on each patient's situation and ensure compliance with governmental consent laws and sub-disciplines regulations.

Our consent enforcement system, shown in Fig. 1 consists of (1) User Interface (UI) for EMR Operations; (2) EMR's Runtime System; (3) Workflow Management System -- a runtime system that enforces medical treatment workflow and checks for consents before enabling a workflow; (4) A Consent Management System that ascertains which consents, if any, are missing and must be issued; (5) A Consent Rule Management System – a system connects to an ontology application and the Consent Service to obtain the appropriate informed consent automatically; and (6) Related Databases. See, the high-level architecture shown in Fig. 3.

Our implementation uses an open source EMR system, OpenMRS [40], and a workflow system YAWL [41]. In our implementation, the EMR user community interacts with the EMR using the well-designed OpenMRS user interfaces. All patient data is stored in OpenMRS' databases. Whenever a treatment procedure (a task to the WfMS) requires a patient's informed consent to move to the next stage, WfMS will call the consent service to retrieve or obtain related consents as a prerequisite to proceeding with the treatment. Patient consents are stored in the OpenMRS' databases as part of their medical records. Consent Management Service is plugged in YAWL as a custom service.

As stated, we enforce medical workflows upon the OpenMRS EMRs by using the YAWL workflow management system. We did so because, first, YAWL workflow system has been used to implement many workflows in industry and academia [42]. Second, YAWL uses a domain independent syntax to specify workflows, and provides an editor and a runtime engine that can enforce workflows specified in YAWL syntax for any applications. Therefore, our models can be audited and verified by third-parties for workflow accuracy. Third, YAWL is open source software. Last, many research projects have recently used YAWL as a workflow-modeling tool. Our medical workflow system is implemented as a loadable module in OpenMRS and incorporates the knowledge of the treatment processes as a YAWL specification. The YAWL workflow engine uses these specifications to provide the caregivers the ability to step through the tasks. In addition, the workflow engine logs every incident into a database creating the audit-able record of the work process provided by

Fig. 3. High level view of workflow based EMRs with consent management


Consent Rule Management System
OWL API
Ontology of Consent Rules
Consents Management System
constraints on the careflow X = f(x₁,x₂,...,xₙ)
Workflow-based EMRs
Modeling Medical Workflows
YAWL Editor
YWAL WfMS
OpenMRS – YAWL Interface
User — EMRs Interface
Admin
Users
OpenMRS

the medical organizations. In another hand, the Consent Management System acts as a customized workflow service in YAWL.

1. *OpenMRS -> YAWL:* ***(Step 1)*** - When a caregiver starts a medical treatment procedure in OpenMRS, a "launch case" event request with workflow specification id or name is sent to YAWL engine; YAWL engine enables some work item(s); *If the enabled work item(s) does not request Consent Service, Then* ***(Step 6)*** - OpenMRS checks out the enabled work item(s) and executes them.

2. *YAWL* enables other appropriate work items based on control flow defined in the workflow specification, sends notification to OpenMRS. Then the interactions between YAWL and OpenMRS are repeated. Otherwise,

3. *YAWL -> Consent Management Service (CMS):* ***(Step2)*** **–** If a task needs to check patient's informed consent, the consent management service is triggered.

4. *CMS -> Ontology Service (OS)* ***(Step 3)****:* CMS uses OWL API to connect to the OS with patient's information and other required consent information. An individual has be created and can be used Pellet to reason appropriate outcomes.

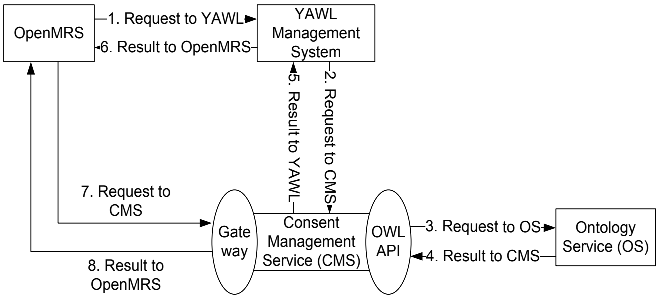
5. *OS -> CMS* ***(Step 4)***: OS retunes the results reasoned based on the SWRL rules to CMS.

6. *CMS -> YAWL* ***(Step 5)***: CMS passed results to YAWL, if valid consents have been hold, obtaining consent from patients medical recodes; otherwise, asks OpenMRS ***(Step 6)*** retrieve appropriate consent forms based on specific treatment task requirements.

7. *OpenMRS -> CMS* ***(Step 7)***: This is additional step existing only required CMS. Asking what kind of consents should be issued.

8. *OpenMRS -> CMS* ***(Step 8)***: Same as the previous step, this is additional step existing only required CMS. CMS return the answers to OpenMRS. The WfMS decides whether the treatment should continue or be aborted based on the treatment specification and on the patient's treatment decision.

Fig. 4. Interactions between the system components



Finally, we pay attention to the privacy and security issues, which are important considerations for any EMRs.

Access Control: The medical team as a whole provides the required services to a patient who visits the medical center, from acceptance of a patient to the end of the treatment at the facility. Each team member plays a designated role in providing care with a set of assigned duties that are choreographed with each other, forming workflows. The team together provides the care planned for the patient. We used a role-based access control model to provide confidentiality. Furthermore, enforced informed consent is an access control with more complex rules.

Accountability: To monitor quality of care and consistent with continuous improvement, an EMR system must have auditing capabilities. In our workflow-enforced EMR system with consent management, the quality care team can review both procedures and outcomes from workflow logs and consent logs, which provide an audit trail that satisfies accountability requirements.

V. Conclusions

Enforcing diverse consent laws in an EMR system is useful for any and all EMR systems, but especially for EMR systems that treat mobile populations, such as military personnel and dependents. We have described an architecture and a prototype system that is based on an open source EMR system, a generic workflow engine and an Ontological rule system. Our system enforces consents for medical treatments, which when deployed will reduce medical malpractice, potential medical treatment errors caused by missing informed consents, and improve the patient-caregiver relationship. The processes of obtaining the consent and including exception processes are also be recorded in the workflow management system, thus becoming available for quality of care audits and reviews.

REFERENCES

[1] E. Coiera, and R. Clark, “e-Consent: The design and Implementation of consumer consent mechanisms in an electronic environment,” Journal of the American Informatics Association (JAMIA) Vol. 11, No 2, Mar/April 2004.

[2] VHA HANDBOOK 1004.05, Transmittal Sheet, (2005, March). “IMEDCONSENT™”. Available at: http://www.va.gov/vhapublications/ViewPublication.asp?pub_ID=1857 Accessed August, 2013

[3] P.S. Appelbaum, and G. Thomas, “The MacArthur Treatment Competence Study. I: Mental illness and competence to consent to treatment”, Law and human behavior 19.2; 1995: 105.

[4] B. Yu, & D. Wijesekera, “Building Dialysis Workflows into EMRs”, HCIST 2013 - International Conference on Health and Social Care Information Systems and Technologies.

[5] B. Yu,, D. Wijesekera, & C. Paulo, “Consent-Based Workflow Control in EMRs”, HCIST 2014 - International Conference on Health and Social Care Information Systems and Technologies.

[6] S. Salgo v. Leland, Jr. University Board of Trustees, 317 P.2d 170-181 (Cal. App. Ct. 1957).

[7] American Medical Association. Professional Resources (Legal Issues) Informed Consent. Available at: http://www.ama-assn.org/ama/pub/category/4608.html Accessed on March 15, 2008

[8] E. Coiera, R. Clark, “e-Consent: The design and Implementation of consumer consent mechanisms in an electronic environment,” Journal of the American Informatics Association (JAMIA) Vol. 11, No 2, Mar/April 2004

[9] C. Ruan, S. S. Yeo, “Modeling of an Intelligent e-Consent System in a Healthcare Domain,” J. UCS, 15(12), 2009. 2429-2444.

[10] G. Russello, C. Dong, and N. Dulay, “Consent-based workflows for healthcare management,” In Policies for Distributed Systems and Networks, 2008. POLICY 2008. IEEE Workshop on (pp. 153-161). IEEE.

[11] C. M. O’Keefe, P. Greenfield, and A. Goodchild, “A decentralized approach to electronic consent and health information access control,” Journal of Research and Practice in Information Technology, vol. 37, no. 2, pp. 161–178, 2005.

[12] J. Bergmann, O. J. Bott, D. P. Pretschner, and R. Haux, “An econsent-based shared EHR system architecture for integrated healthcare networks,” International Journal of Medical Informatics, vol. 76, pp.130–136, 2007.

[13] K. T. Win, J. A. Fulcher, “Consent mechanisms for electronic health record systems: A simple yet unresolved issue,” Journal of Medical Systems, vol. 31, pp. 91–96, 2007.

[14] N. P. Sheppard, R. Safavi-Naini, and M. Jafari, “A digital rights management model for healthcare,” In Policies for Distributed Systems and Networks, 2009. POLICY 2009. IEEE International Symposium on (pp. 106-109). IEEE.

[15] K. Win, H. Song, P. Croll, and J. Cooper, “Implementing patients consent in electronic health record systems,” Proceedings of CollECTeR, Melbourne, Australia, 2002.

[16] J. Grimson, G. Stephens, B. Jung, W. Grimson, D. Berry, S. Pardon, “Sharing health-care records over the internet,” IEEE Internet Comput. 5 (3) (2001) 49–58.

[17] N. Saranummi, “PICNIC architecture,” Studies in health technology and informatics, 115, 37-60. 2

[18] Web service semantics in the eHealth Domain: The Artemis Project, Available from: http://www.metu.edu.tr/search-page?cx=011418946324636299173%3Ajgqmx6nposk&ie=utf-8&qa=ARTEMIS%20project%20 Access on January 10, 2014

[19] C. Rosse, JLV. Mejino, “A reference ontology for biomedical informatics: the Foundational Model of Anatomy,” J Biomed Inform 2003;36:478–500.

[20] AC. Yu, “Methods in biomedical ontology,” J Biomed Inform 2006;39:252–66.

[21] J. Dang, A. Hedayati, K. Hampel, C. Toklu. An ontological knowledge framework for adaptive medical workflow. J Biomed Inform 2008;41:829–36.

[22] E. E. Matos, F. Campos, R. Braga, D. Palazzi, “CelOWS: an ontology based framework for the provision of semantic web services related to biological models,” J. Biomed Inform 2010;43:125–38.

[23] A. Valls, K. Gibert, D. Sánchez, M. Batet, “Using ontologies for structuring organizational knowledge in home care assistance,” Int J Med Inform 2010;79:370–87.

[24] D. Riaño, F. Real, J. A. López-Vallverdú, F. Campana, S. Ercolani, P. Mecocci, and C. Caltagirone, “An ontology-based personalization of health-care knowledge to support clinical decisions for chronically ill patients,” Journal of biomedical informatics, 45(3), 429-446.

[25] “BioPAX : Biological Pathways Exchange,” Available at: http://www.biopax.org /Accessed on July 18, 2014

[26] "About CCO and GexKB", Available at: http://www.semantic-systems-biology.org/apo/ Access on December 3, 2013

[27] J. D. Osborne, J. Flatow, M. Holko, S. M. Lin, W. A. Kibbe, L. J. Zhu, and R. L. Chisholm, “Annotating the human genome with Disease Ontology,” BMC genomics, 10(Suppl 1), S6, 2009.

[28] M. Van Gurp, M. Decoene, M. Holvoet, M., and M. C. dos Santos, “LinKBase, a Philosophically-Inspired Ontology for NLP/NLU Applications,” In KR-MED, November, 2006.

[29] P. L. Whetzel, and et al. “NCBO Technology: Powering semantically aware applications,” J. Biomedical Semantics, 4(S-1), S8, 2013.

[30] W. J. Bug, Ascoli, J. SGrethe, A. Gupta, C. Fennema-Notestine, A. R. Laird, A. R., ... and M. E. Martone, ”The NIFSTD and BIRNLex vocabularies: building comprehensive ontologies for neuroscience,” Neuroinformatics, 6(3), 175-194, 2008.

[31] M. Q. Stearns, C. Price, K. A., Spackman, and A. Y. Wang, “SNOMED clinical terms: overview of the development process and project status,” In Proceedings of the AMIA Symposium (p. 662). American Medical Informatics Association. 2001.

[32] B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters,... and S. Lewis, “The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration,” Nature biotechnology, 25(11), 1251-1255. 2007

[33] J. Day-Richter, M. A., Harris, M. Haendel, and S. Lewis,”OBO-Edit—an ontology editor for biologists.” Bioinformatics, 23(16), 2198-2200. 2007

[34] D. A. Natale, C. N. Arighi, W. C, Barker, J. A. Blake, C. J. Bult, M. Caudy,... and C. H. Wu, “The Protein Ontology: a structured representation of protein forms and complexes,” Nucleic acids research, 39(suppl 1), D539-D545. 2011.

[35] T. L. Beauchamp, J. F. Childress. “Principles of Biomedical Ethics,” Third Edition. New York: Oxford University Press, 1989:1–470.

[36] R. R. Faden, C. Becker, C. Lewis, J. Freeman, and A. I. Faden, “Disclosure of information to patients in medical care,” Medical Care, 718-733.1981.

[37] D. M. Studdert, M. M. Mello, M. K., Levy, R. L. Gruen, E. J. Dunn, E. J. Orav, and T. A. Brennan, “Geographic variation in informed consent law: two standards for disclosure of treatment risks,” Journal of Empirical Legal Studies,4(1), 103-124, 2007.

[38] C. B. Fisher,“Goodness-of-Fit Ethic for Informed Consent”, A. Fordham Urb. LJ, 30, 159, 2002

[39] F. C. Bourgeois, P. L. Taylor, S. J. Emans, D. J. Nigrin, and K. D. Mandl, “Whose personal control? Creating private, personally controlled health records for pediatric and adolescent patients,” Journal of the American Medical Informatics Association, 15(6), 2008. 737-743.

[40] “OpenMRS Developer Guide”. OpenMRS Website, Available at: https://wiki.openmrs.org/display/docs/Developer+Guide Accessed May, 2013.

[41] “YAWL Technical Manual 2.1 version”. YAWL Website, Available at: http://www.yawlfoundation.org/manuals/YAWLTechnicalManual2.1.pdf Accessed May, 2013.

[42] “YAWL User Manual”. *YAWL Website*, Available at: http://www.yawlfoundation.org/yawldocs/YAWLUserManual2.0.pdf Accessed May, 2013.

Processing Events in Probabilistic Risk Assessment

Robert C. Schrag, Edward J. Wright, Robert S. Kerr, Bryan S. Ware
Haystax Technology
McLean, VA USA

Abstract—**Assessing entity (e.g., person) risk from entity-related events requires appropriate techniques to address the relevance of events (individually and/or in aggregate) relative to a prevailing temporal frame of reference—for continuous risk monitoring, a running time point representing "the present." We describe two classes of temporal relevance techniques we have used towards insider threat detection in probabilistic risk models based on Bayesian networks. One class of techniques is appropriate when a generic person Bayesian network is extended with a new random variable for each relevant event—practical when events of concern are infrequent and we expect their number per person to be small (as in public records monitoring). Another class is needed when (as in computer network event monitoring) we expect too many relevant events to create a new random variable for each event. We present a use case employing both classes of techniques and discuss their relative strengths and weaknesses. Finally, we describe the semantic technology framework supporting this work.**

Index Terms—temporal relevance; event relevance; anomaly detection; qualitative Bayesian network specification; probabilistic model; insider threat

I. Introduction

Different parties have legitimate interests in understanding the risks that may be incurred when given persons are allowed to act in given roles. Employers are concerned about prospective employees, lenders about borrowers, landlords about tenants, and judges and parole boards about convicted criminals. To each role is accorded some privilege or stake—access to information/influence/reputation, finance, property, or liberty/public safety—that a bad actor could abuse or damage. While it's usually impossible to predict how a specific person *P* may behave in a given role *R*, an interested party *Q* may apply a probabilistic risk model *M* to available information about *P* to understand where *P*'s assessed risk may fall relative to other current or prospective players of *R*. *M* may:

- Have been derived from similar available data about other persons considered or selected for *R*
- Be based on legal or other policy doctrine
- Embody knowledge elicited from subject matter experts or published in a theory of human psychology.

Depending on the outcome of *M* applied to *P, Q* may choose to:

- Engage *P* in *R* (or not)
- Modify or disengage from its *R* relationship with *P*
- Invest more resources in assessing *P*'s risk (perhaps monitoring *P*'s actions)
- Counsel *P* in the positive management of factors related to *P*'s risk
- Modify *M* to accommodate an acceptable pattern of behavior not earlier addressed.

We have developed a series of related person risk models addressing the risk that *P* poses to *Q*. Each interprets the set of known events *E* involving *P* in assessing *P*'s risk. Each also must address the relevance of a given event *e* in *E* to computing *P*'s risk at a given time point *t* in *T*—the entire interval of relevant events (beginning, e.g., at *P*'s birth, majority, or engagement with *Q* and ending at the present or a most recently available event report date). Each model is probabilistic, calculating its core risk assessment using a Bayesian network (BN) [3]. Each model includes a generic person BN $\mathscr{B}$, which it may extend (based on *P*'s events) to create a person-specific BN $\mathscr{B}_P$.

A. Model $\mathrm{M_C}$: *Processing* P'*s Life Events with Ingestion Rules*

M_C addresses the risk that *P* may disclose *Q's* private information without proper authorization, considering relevant event types (say, technical certification or conduct reprimand) that may collectively have a few instances per year. To address the effect of *P*'s events *E* on *P*'s risk, M_C runs a set of "ingestion" rules, each of which may extend $\mathscr{B}$ to reflect a given event *e* in *E,* ultimately resulting in $\mathscr{B}_P$. Because each triggered ingestion rule adds one or more random variables to $\mathscr{B}$, this approach tends to be practical when *E* is small relative to $\mathscr{B}$ (so that $\mathscr{B}_P$ does not grossly exceed expected computational requirements). M_C addresses the temporal relevance of a given event *e* by arranging that *e*'s influence on risk will build (when *e* is ongoing at *t*) or decay (when *e* is completed at *t*). This mechanism creates a conditional probability table associated with the temporal relevance of *e* to reflect the juxtaposition of *t* (the reference time for this risk calculation) with respect to the time point at which *e* occurs (if *e* is a point event) or the respective time points at which *e* begins and ends (if *e* is a durative event).

B. Model $\mathrm{M_S}$: *Processing* P'*s Network Events with Summary Random Variables*

M_S addresses the risk that *P* may pose an insider threat to *Q* via its access to *Q*'s information technology (IT) system—*Q*'s computers, computer networks, and related assets. The threat may be unauthorized information access, disclosure, theft, or destruction. M_S considers relevant event types (e.g., copying a file to a thumb drive or to an external website) that may occur many times per day. Given *Q*'s interest in assessing *P*'s risk on a continuous basis—over an employment interval—the "ingestion" approach that M_C uses to grow $\mathscr{B}_P$ with every event (instance) *e* in *E* is not feasible. Instead, for each such fine-grained event type *τ,* we include in M_S's version of $\mathscr{B}$ a random variable (RV) summarizing the extent to which *P*'s actions are believed to warrant a suspicion of *P*'s exploiting *Q*'s IT assets towards insider threat. We calculate a likelihood for this summary RV so as to reflect:

- The relative novelty or familiarity of *P*'s events of type τ over:
  - *P*'s history
  - The synchronous history of other persons playing role *R* for *Q*
- Temporal decay.

M_S also considers relevant event types (e.g., copying a file to Wikileaks) that obviously manifest (vs. just warrant suspicion regarding) insider threat. For these latter event types, M_S uses the same ingestion approach as M_C.

C. Model $\mathrm{M_G}$: *Combining* $\mathrm{M_C}$ *and* $\mathrm{M_S}$

M_G combines M_C and M_S to address both the above aspects of insider threat—so that each model aspect can inform the other. E.g., if *P*'s non-network life events lead (the M_G version of) $\mathscr{B}_P$ to believe that *P* is likely **`Untrustworthy`**, this will increase (relative to a baseline, skeptical model state) $\mathscr{B}_P$'s belief that any potentially suspicious computer network actions actually do warrant suspicion. Also, staff members who warrant more insider threat suspicion on the computer network side receive higher overall risk scores, facilitating *Q*'s ability to react appropriately in general.

D. Common Modeling Framework

We have (following [5] and [6]) developed the generic person BNs $\mathscr{B}$ for the above models in our generic framework for large-scale probabilistic modeling that lets us specify:

- Boolean-valued RVs (generally, person attribute concepts—e.g., **`Trustworthy`**)
- Directed influences between concepts with discrete, qualitative strengths (obviating the usual BN requirement to specify—manually—for each RV a conditional probability table with one numeric entry for each element in the Cartesian product of its parents' domains—i.e., 2^n for *n* Boolean-valued domains)
- Generic modeling patterns for concept indication, mitigation, and relevance.

Our framework compiles such qualitative specifications into a representation executable by an off-the-shelf BN tool. (We use the Netica® API from Norsys.) Our $\mathscr{B}$ for M_C includes hundreds of RVs. Our specification of $\mathscr{B}$ for M_S is small enough to exhibit below (see Fig. 10, in the Appendix). $\mathscr{B}$ for M_G is again large—and the framework's support for layering of qualitative source specifications affords a much easier path to M_G than if we had built an M_S BN outside the framework.

Our ingestion rules that extend the generic person $\mathscr{B}$ into a person-specific $\mathscr{B}_P$ are described further in section VI.

E. Sequel

The sequel describes techniques we use to implement temporal relevance under the ingestion and summary approaches (introduced above with M_C and with M_S, respectively). We also exhibit results from the combined model, M_G, and discuss design trade-offs. Finally, we describe our supporting semantic technology framework.

II. Computing Temporal Relevance for Events Individually Ingested

Intuitively, the impact of a major life event on one's reputation is time-limited. While positive life events tend to build our confidence in a person—and negative ones erode it—the glow of accomplishment—like the stain of failure or breach—naturally fades over time. In our "whole-person" model M_C, we uniformly invoke exponential decay (or growth) with half life γ per an invoked ingestion rule Γ. The generic person Bayesian network (BN) $\mathscr{B}$ accounts for interactions among beliefs about random variables (RVs) representing different person attribute concepts like those in Fig. 1.

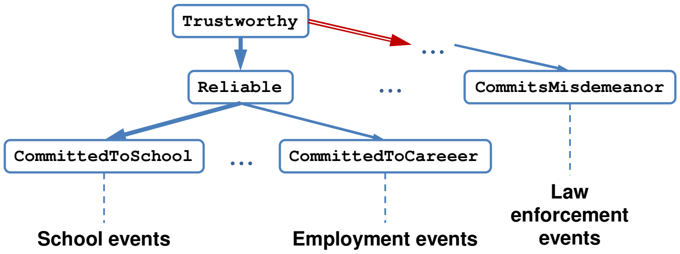


Fig. 1. Partial generic person attribute concept BN $\mathscr{B}$ (top), with related event categories (bottom).

In Fig. 1, BN influences point (causally) from indicated concept hypothesis to indicating concept. Stronger indications have thicker arrows, a single negative indication has a red, double-lined arrow. The full BN includes several hundred nodes (mostly elided).

M_C's ingestion rules apply *P*'s event evidence to $\mathscr{B}$ to develop a person-specific BN $\mathscr{B}_P$ including temporal relevance RVs (as explained next) appropriate for a given reference time point *t*. $\mathscr{B}_P$ then calculates the risk at *t*. By constructing $\mathscr{B}_P$ and calculating risk at successive time points, we develop a historical risk profile (i.e., a risk timeline) for *P*. See Fig. 2.

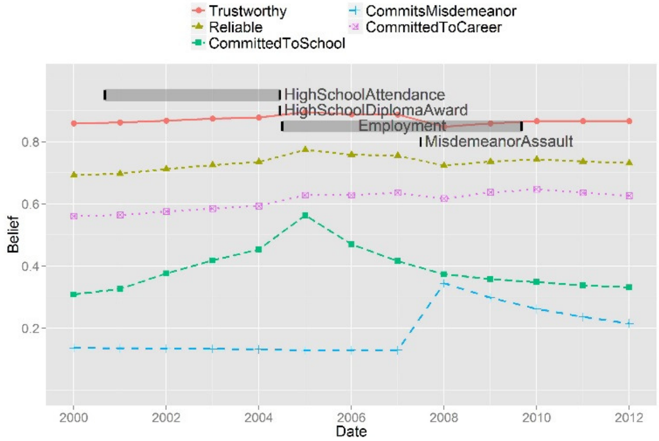


Fig. 2. Person risk timeline with life event overlay—per M_C.

In Fig. 2, *P*'s events are plotted in bars (top left to center). Belief over time is plotted for person attribute concept RVs per legend (top). **`Trustworthy`** is our top-level proxy for (the complement of) risk. Note how beliefs in **`CommittedToSchool`** and **`CommittedToCareer`** tend to build while the related (**`HighSchoolAttendance`** and **`Employment`**) events are ongoing. Influence interactions in $\mathscr{B}$

cause belief in `CommittedToCareer` to grow even while P is still in high school. (We tend to believe that someone who does well in school will also do well in a career.) Belief in `CommittedToSchool` increases when P graduates but then become less relevant per half lives specified in ingestion rules for school-related events. The 2007 `MisdemeanorAssault` charge decreases belief in all the other, positive concept RVs. See also Lisp macro calls expressing associated event data in Fig. 9.

When Γ ingests an event e (e.g., of type `EmploymentReprimand`), it instantiates a BN design pattern that:

- Creates an RV δ standing for e itself (an evidence RV)
- Creates an RV ρ standing for the temporal relevance of e
- Installs both ρ and the indicated person attribute RV π (standing for, e.g., `DisregardsEmploymentRules`) as BN parents of δ (see left panel in Fig. 3, below)
- Creates appropriate conditional probability tables (CPTs) for δ and ρ—denoted CPT(δ) and CPT(ρ).

CPT(ρ) encodes e's nominal relevance at t, calculated per ingestion rule Γ's specified half life γ and the time α elapsed from e's time point (designated by Γ as "beginning" or "ending," when e is durative) until the reference point t. For the case of relevance decay, we have $\theta = ½^{(\alpha / \gamma)}$. We specify θ as the probability P(ρ = *"true"*) and $1 - \theta$ as P(ρ = *"false"*).

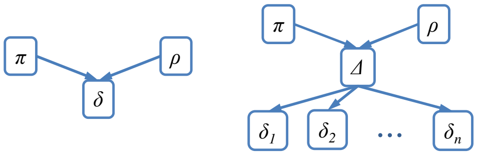


Fig. 3. Current ingestion scheme (left) and potential variant (right).

In Fig. 3 (left), BN influences are associated with an (ingested event) evidence random variable (RV) δ, an indicated person attribute concept π, and a temporal relevance RV ρ. In Fig. 3 (right), an evidence summarization scheme (using summary RV Δ) can insulate similarly-typed, closely temporally spaced events δ_i against departures from their nominally specified temporal relevance beliefs (say, θ_i) that otherwise would be induced among individual RVs (say, ρ_i)—had rather these been used.

CPT(δ) respects e's strength (specified in Γ) as an indicator for π and arranges that the probabilities P(π = *"true"*) and P(π = *"false"*) observed in $\mathscr{B}$ for π do not depart (via the normal course of Bayesian influence propagation) from the nominal value θ installed in CPT(ρ). This is a local correction that is subject to further departures when other ingestion rule executions also modify $\mathscr{B}_P$. While we have observed this ingestion technique for temporal relevance to work well in practice, when two or more events in E are both semantically and temporally close to each other, we again see temporal relevance departures resulting from Bayesian influence propagation in $\mathscr{B}_P$. (The relevance RVs ρ_i tend to reinforce each other, amplifying their observed beliefs beyond their nominal θ_j. In some applications, this pattern may be appropriate; in others not.) We can ingest two nearly simultaneous (like-type) misdemeanor events without blatant departure from nominal θ. Ingesting five such events, we see θ decay only some 6% γ days after the events' occurrence (when we might naively have expected 50%).

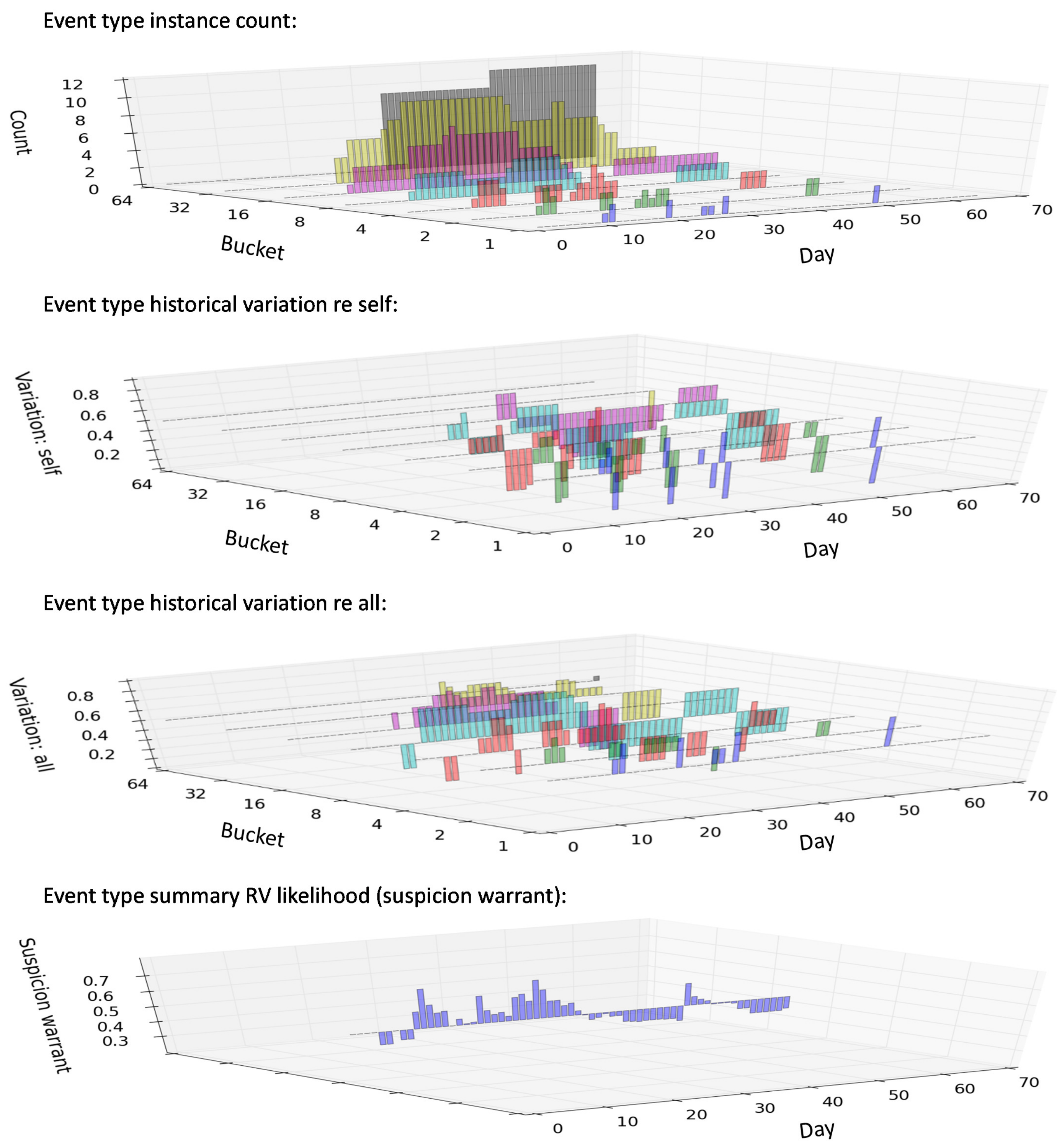
We can—for quasi-simultaneous events—decouple the influence of temporal relevance from multiple indicating evidence events by invoking the alternative BN design pattern in in Fig. 3 (right panel), where Δ is a summary RV for individual event RVs δ_1, δ_2, … , δ_n. Accommodating evidence events δ_i occurring at materially different time points requires a more general approach to avoid the departures of temporal relevance beliefs from nominally specified values. The approach we describe in section III works well in this regard, but it does not afford the same expressive power as ingestion rules (which can consider arbitrary temporal relationships between events—as discussed in section V).

III. Temporal Relevance with Summarized Events

Computer network events that may inform Q about an insider threat by its engaged staff member P can occur so frequently that the ingestion technique described in section II is impractical. M_S avoids this issue by appealing to event summary RVs, as outlined in section I. See Fig. 10 in the Appendix, where the RV `CopyDecoyToExternal_Summary` (e.g.) summarizes the suspiciousness of actions in which P has committed the network action `CopyDecoyToExternal` (i.e., copy a seeded "decoy" file to an external location, such as a website).

Fig. 4 (full page) exhibits key metrics we compute for such a summary RV. Because we expect network event monitoring to be continuous—with practically unbounded beginning and no ending—we compute key metrics in temporal buckets of exponentially increasing size (top three charts)—so that we can always double temporal (if not event) capacity by adding one more bucket. Event occurrence buckets summarize (top/first) event type count since monitoring started, (second) signal with respect to P's own history, (third) signal with respect to (a statistic computed over) the full/relevant monitored population. Each of 64 day's variation metrics (middle charts) are computed with respect to just the other buckets for that day, normalizing ratios of counts in related buckets to the range [0, 1] using a sigmoid function. Visual "floors" in the bottom three charts are set at 0.5 (the sigmoid function's point of symmetry)—highlighting direction of signal change. Variation with respect to own history compares events new in a bucket (those counted in the next-largest bucket) to those that are old—reverting to bucket-dependent defaults when no earlier events exist. To develop a suspicion likelihood (bottom chart), we first take a weighted average of each of the two anomaly metrics (with weights increasing, e.g., by bucket recency), then average these results. We then enter this suspicion likelihood (using the Netica® API) as a likelihood finding on a summary RV (such as `CopyDecoyToExternal_Summary`).

Under even weighting (invoked in Fig. 4 and Fig. 5), e's relevance approximates $1/\alpha$. Compare this to our half life decay function (from section II) used under ingestion: $½^{(\alpha / \gamma)}$. Either class of techniques would in fact be compatible with either of these (or other) functions of relevance over time. We reviewed the overall half life approach and half lives appropriate for specific event types in M_C with experts in the subject matter of unauthorized information disclosure risk. Decay rates for computer network events in M_S have yet to be tuned in the context of real-world data.


Event type instance count:

Event type historical variation re self:

Event type historical variation re all:

Event type summary RV likelihood (suspicion warrant):

Fig. 4. Key metrics for a summary RV in an overall event type-related suspicion warrant.

Not yet exhibited in our processing in Fig. 4 and Fig. 5 is an approximating space optimization that would shift the "old" content (see Fig. 4) from bucket *j* into bucket *j* + 1 at each $(2^j)^{th}$ time step (for values *j* descending from the highest value for which ($t \bmod 2^j$) = 0—rather than computing the buckets' counts afresh at each time step, as shown. The latter approach would require retaining full event counts for all time steps—impractical for long-term, continuous operation.

A. Related Work

Senator et al. [4] describe a flexible insider threat detection framework providing statistical and machine learning components that may be applied to data across different time scales. In contrast to our bucketed approach, their time scales must be explicitly specified by an application architect. They illustrate a component workflow motivated by a specific threat scenario. We have taken such scenarios to be compiled into random variables (for indicators and threats) in a Bayesian network (focused on overall risk assessment, rather than specifically on threat incident detection). Our framework can address a broad set of statistics in and over temporal buckets, supporting event processing over an arbitrary time scale. They describe results with a real-world dataset covering two months. We have developed our approach using a similar synthetic dataset [1] covering 18 months.

IV. Combined Model Use Case

We combine M_C with M_S—producing M_G—by appending the input models' influence graph specifications and defining M_S's **`Untrustworthy`** as the opposite of M_C's **`Trustworthy`**. As discussed in section II, this affords a path for *P*'s non-network, life events to influence the risk measured for *P*'s network events—thus enhancing the signal to noise ratio for persons who seem risky generally. See Fig. 5.

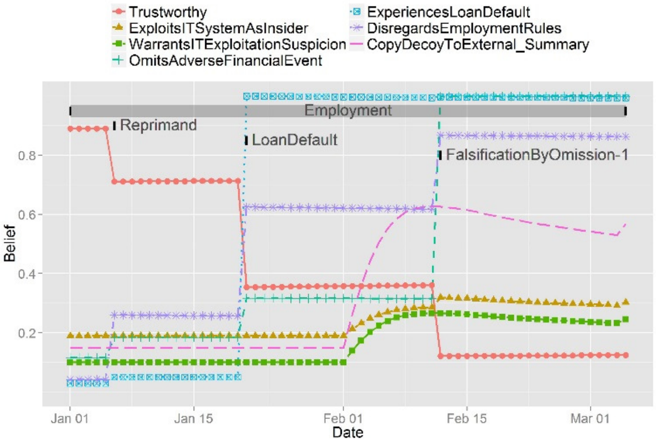


Fig. 5. Person risk timeline—per M_G.

In Fig. 5, life events (top, left to center) are per the M_C component. Belief in *P*'s attribute concept RV **`Trustworthy`**—our top-level proxy for (the complement of) risk—falls lower with each successive derogatory life event (**`Reprimand`**, **`LoanDefault`**, **`FalsificationByOmission`**). This increases the relevance of **`WarrantsITExploitationSuspicion`** to **`ExploitsITSystemAsInsider`** in the M_S component. (This influence is not mitigated, because *P* is not engaged by *Q* in the role of IT administrator—see Fig. 10 in the Appendix.) Belief in **`WarrantsITExploitationSuspicion`** (third belief line from bottom on plot's right) takes a jump of about 10% when M_S's summarized events mount (level until day 33, then increasing by one counted event per day) and M_C's ingested events have occurred. By comparison (not shown), belief in **`WarrantsITExploitationSuspicion`** jumps by only about 3% when M_S stands alone, uninformed by an M_C component. Life events decay per half lives not conspicuous at the depicted time scale.

M_G is a proof of concept. Additional cross-model linkage and tuning of relevant modeling parameters may further increase the signal to noise ratio for network user risk detection when non-network, life events are consider in the same unified model.

V. Discussion: Event Processing Design Trade-Offs

The different domains we have addressed in M_C and M_S have presented event processing requirements largely amenable to—in fact, engendering—the two classes of techniques described here: per-event ingestion (section II) and event summarization (section III), respectively. As noted in sections II and III, per-event ingestion is liable to (possibly unintended) amplifications of temporal relevance, when event instances are both semantically and temporally close. This is, however, just the situation for which we have designed event summarization. While Fig. 3 (right) suggests a hybrid approach for M_C to aggregate events that are temporally close, this really begs the question: *What should be the effect of similar indicating events on the belief calculated for an indicated hypothesis random variable (RV) in a Bayesian network (BN) $\mathscr{B}$?*

Event summarization, in M_S, adopts the extreme position that all events of a given type τ should be summarized in a single random variable (RV). The uniformity of this approach may be appealing, but it bears a simplicity driven by the necessity of addressing a practically unlimited stream of fine-grained events—many of which are relatively weak indicators of insider threat. Event ingestion, in M_C, hazards (so far, in our application domain, rare) potential amplifications in temporal relevance but affords the power (via Allegro Prolog®-based ingestion rules and auxiliary predicates—see section VI.A) to express nuanced temporal configurations of events of different types that are not obviously amenable to bucketed historic summaries. We might note temporal overlaps to extract certain compound events (say, of type **`FailsDrugTestDuringEmployment`**), but we could not refer to earlier relevant events, as in Fig. 6.

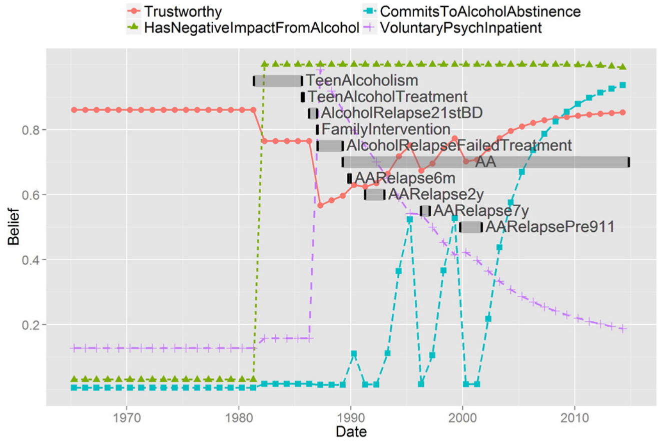


Fig. 6. Ingestion rule processing of non-overlapping events in M_C.

In Fig. 6, *P*, with a history of alcoholism, has an ongoing commitment to alcohol abstinence but also several intervening alcohol problem events (relapses). While M_C considers an alcohol abstinence commitment to mitigate an earlier alcohol problem, this commitment is voided (and the building of *P*'s credibility begins again) when *P* relapses. In M_C, the temporal specificity of a mitigating event (or generality of a mitigating person attribute concept) is important in determining whether to capture this effect with an ingestion rule, with a **`:MitigatedBy`** influence specification (see Fig. 10 in the Appendix), or with a combination of such mechanisms. M_C's related ingestion rule invokes our Allegro Prolog® predicate **mostRecentLaterStartingReportedEvent** to rebase the temporal relevance computation for the person attribute concept **CommitsToAlcoholAbstinence** at a most recent relapse event's ending point.

Intimately related to realizing an appropriate overall semantics for a person risk rating model M (but outside the scope of the present discussion about processing person events) is the design of person attribute concept RVs and concept-to-concept influences in a generic person BN $\mathscr{B}$. Even before we decide how to associate encountered evidence E with $\mathscr{B}$, we must be happy with $\mathscr{B}$'s inferences under arbitrary (likelihood or domain value) findings for $\mathscr{B}$'s RVs. This requires thinking (and testing) at least as hard about the semantic relationships among person attribute concept definitions and connecting influences as we do about those among event types and processing styles. (See also section VI.C.)

VI. Supporting Semantic Technology

Our ingestion rules are written in Allegro Prolog®. They read events expressed using an OWL ontology from an AllegroGraph® triple store and create a person-specific BN $\mathscr{B}_P$ using the Allegro Common Lisp® API to the Netica® API. Allegro Prolog®, AllegroGraph®, Allegro Common Lisp® are products of Franz, Inc. Netica® is a product of Norsys, Inc. The Allegro Common Lisp® API to the Netica® API is open-source. We see unique benefits in this software stack.

A. Ingestion Rule Design

AllegroGraph® is an RDF triple store management system that happens to be written in Allegro Common Lisp®. While Franz supports AllegroGraph® clients for a number of different languages, the direct (vs. remote) Lisp client benefits us in that it shares memory with AllegroGraph® itself. Allegro Prolog®, written in and included in Allegro Common Lisp®, is a logic programming facility that the Lisp direct client extends with Lisp macros and Prolog predicates affording access (alternatively to SPARQL) to AllegroGraph® triple stores. Because Allegro Prolog® supports calls to Lisp functions from within logic programming rules, our ingestion rules can invoke the Allegro Common Lisp® API to the Netica® API to augment an existing generic person Bayesian network (BN) model $\mathscr{B}$ to add random variables (RVs) corresponding to a person P's events E, resulting in a person-specific BN $\mathscr{B}_P$. See Fig. 7.

```
(defIngestionRule RestrainingOrder
     (+process-reportedEvent ?person ?*asOfDate)
  (reportedEvent ?person
                 ?*asOfDate
                 ?event
                 !agent:ProtectiveRestrainingOrder
                 ?*startDate
                 ?*endDate
                 ?*ongoing?
                 ?*reportDate)
  (lisp (create-EventConceptIndication
         ?person
         :IndicatedConcept CommitsDomesticViolence
         :+IndicatingEvent ?event
         :Terminus :end
         :DeltaDays (- ?*asOfDate ?*endDate)
         :HalfLife (* 6 365)
         :Strength :strong
         :Polarity :positive)))
```

Fig. 7. M_C ingestion rule.

In Fig. 7, **RestrainingOrder** names an ingestion rule pertaining to events of type **https://haystax.com/agent#ProtectiveRestrainingOrder** (whose RDF namespace part is signaled in AllegroGraph® by the prefix **!agent:**). **defIngestionRule** is a macro wrapping Allegro Prolog® **<-**, registering the ingestion rule and performing static analysis to ensure well-formedness. **+process-reportedEvent** is the predicate—of which all ingestion rules are members—used to launch ingestion rules for a given person and reference time. Logic programming variables are prefixed by **?**, Common Lisp keywords by **:**. We use the prefix **?*** as a convention noting that a binding should have a native Lisp value, rather than an RDF part (i.e., a resource or a literal). **?person** and **?asOfDate** will be bound when the rule is called. The call to **reportedEvent** succeeds when **?event** can be bound to an instance of **!agent:ProtectiveRestrainingOrder**, such that **?person** is the **!agent:riskRatingSubject** of **?event**, and **?event**'s temporal characteristics and provenance are appropriate (binding values for other logic variables). When **reportedEvent** succeeds, the call to the Lisp function **create-EventConceptIndication** is executed, augmenting $\mathscr{B}_P$ as explained in section II and illustrated in Fig. 3 (left): **CommitsDomesticViolence** takes the role of ρ, **?event** induces the new random variables δ and π, the value of the **:DeltaDays** keyword argument takes the role of α, that of the **:HalfLife** argument the role of γ. The ingestion rule itself serves as Γ. Upon completed ingestion processing, the realized $\mathscr{B}_P$ can be compiled and queried for beliefs in person attribute concepts of interest.

Allegro Prolog® includes predicate-level functors supporting logical operations (e.g., **and**, **or**), backtracking control (varieties of **if**, cut), and Lisp calls evaluated at predicate level for their truth values (i.e., not just execution for side effect as in Fig. 7). Under AllegroGraph®'s direct Lisp client, user-defined Allegro Prolog® rules (so ingestion rules and their supporting predicates) may include any RDF resources (i.e., URIs) or literals.

B. Event Ontology Design

With its signature treatment of programs as data (both expressed as lists), Lisp has long been a favorite language for creating embedded knowledge representation languages and supporting utilities. We exploit this facility in designing our models' ontologies for person-related events—using Lisp macros to express class, property, and individual (instance) definitions. See Fig. 8.

```
(defOntologyClass Person (Thing)
  (hasGender Gender :Functional))

(defOntologyClass Gender (Thing)
  (:enumeration Male Female OtherGender))

(defOntologyType Date !xsd:date)

(defOntologyClass Event (Thing)
  (riskRatingSubject Person :Functional)
  (startDate Date (:cardinality 1))
  (endDate Date :Functional)
  (sourceReport Report :Functional))

(defOntologyClass PointEvent (Event)
  (hasConsequentEvent Event))

(defOntologyClass DurativeEvent (Event)
  (hasSubEvent Event))

(defOntologyClass ProtectiveRestrainingOrder
    (PointEvent))
```

Fig. 8. Lisp macro calls defining elements of our event ontology for M_C.

Macro calls in Fig. 8 add triples to a specified graph in an active store. Store-resident triples may be serialized to a standard OWL file in (e.g.) RDF/XML format, then viewed in an available ontology browser (e.g., Protégé). RDF namespace designations (e.g., **!xsd:**) are required only where these deviate from a specified default. For a specified class (e.g., **Person**), an object or datatype property (e.g., **hasGender** or **startDate**) is created per the type (e.g., **Gender** or **Date**) specified. OWL closed enumeration classes (e.g., **Gender**) are supported, as are OWL property types (e.g., **Functional**) and restrictions (e.g., **cardinality**). Validation machinery ensures a specified ontology's global consistency with respect to effective cardinalities allowed.

Per Fig. 8, we now have a single-actor event ontology. While M_S defines persons' roles (e.g., system administrator) with respect to organizations, we have not yet broached persons' roles in events beyond the person-indexing property **riskRatingSubject**.

Our framework also has a Lisp macro useful for defining hand-crafted datasets. See Fig. 9.

```
(defOntologyInstance !data:P (Person))

(defOntologyInstance !data:PHighSchoolAttendance
    (SchoolAttendance)
  (riskRatingSubject !data:P)
  (schoolCredentialAward !data:PDiplomaAward)
  (startDate "2000-09-04")
  (endDate "2004-06-15"))

(defOntologyInstance !data:PDiplomaAward
    (SchoolCredentialAward)
  (riskRatingSubject !data:P)
  (startDate "2004-06-15")
  (schoolCredentialAwarded HighSchoolDiploma))

(defOntologyInstance !data:PEmployment
    (Employment)
  (riskRatingSubject !data:P)
  (startDate "2004-07-05")
  (endDate "2009-09-05"))

(defOntologyInstance !data:PMisdemeanorAssault
    (PoliceOffense)
  (riskRatingSubject !data:P)
  (offenseChargeSchedule Misdemeanor)
  (startDate "2007-06-30"))
```

Fig. 9. Lisp macro calls used to create the (minimal) dataset for the person profiled in Fig. 2.

The framework validates any loaded dataset with respect to declared subject and object classes, literal data types, and property types (e.g., **Functional**) and restrictions (e.g., **cardinality**).

C. Probabilistic Ontology Design

We do not now break down person attribute concepts (e.g., **Trustworthy**) beyond their status as such. Conceptually, they are properties of **Person** that—via their corresponding random variable (RVs) in $\mathscr{B}$—constitute a (flat) probabilistic ontology [1]. Relationships among these RVs are of the kind specified in Fig. 10. Most person attribute concept definitions in M_C include citations to and/or excerpts from guiding policy documents regarding information disclosure risk that also specify related indicating, mitigating, and relevance-inducing concepts.

VII. Conclusion

We have described two classes of techniques for processing events in probabilistic person risk models, examining the advantages and disadvantages of techniques in each class. Our proof-of-concept (section IV) combination of techniques from both classes demonstrates how inferences informed by either class of event processing can inform the other effectively. The selection of event processing techniques is one key element of overall risk model design, along with event ontology design and influence network design. In support of this work, we have developed and exploited appropriate semantic technology, with an eye towards flexible reuse.

Disclaimer

The views expressed are those of the authors and do not reflect the official policy or position of any legally recognized body or its representative parts or members.

References

[1] P. Costa and K. Laskey. "PR-OWL: A framework for probabilistic ontologies." Frontiers in Artificial Intelligence and Applications 150 (2006): 237.

[2] J. Glasser and B. Lindauer. "Bridging the gap: A pragmatic approach to generating insider threat data." IEEE Security and Privacy Workshops, 2013.

[3] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Series in Representation and Reasoning, 1988.

[4] T. Senator et al., "Detecting insider threats in a real corporate database of computer usage activity," Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2013.

[5] C. Twardy, E. Wright, S. Canon, and M. Takikawa, "Credibility models," in Proceedings of the 5th Bayesian Modeling Applications Workshop. Vancouver, 2007.

[6] E. Wright and K. Laskey, "Credibility models for multi-source fusion," Proceedings of the 9th International Conference on Information Fusion, July 2006.

Appendix: Qualitative Specification for the Generic Person Bayesian Network in M_S

The generic person Bayesian network (BN) $\mathscr{B}$ for M_S is as specified in Fig. 10. "**_Summary**" random variables (RVs) correspond to an event schema derived from US CERT synthetic dataset "r6.2" available at http://cert.org/insider-threat/tools/index.cfm [1].

In Fig. 10, the top-level RV **ExploitsITSystemAsInsider** is a disjunctive summary of two next-level RVs: **CommitsITExploitation** (*P* has committed an unambiguously exploitative event—presumed rare) and **WarrantsITExploitationSuspicion** (covering network events that may or may not be exploitative). **CommitsITExploitation** is absolutely indicated ("implied") by three intermediate RVs that are in turn indicated (with varying strengths) by computer network event summary RVs. M_S enters likelihood findings for the latter into $\mathscr{B}_P$ as described in section III. **WarrantsITExploitationSuspicion** is considered relevant (absolutely) to **CommitsITExploitation** if *P* is **Untrustworthy**, mitigated (strongly) if *P* **HasRole-ITAdmin** (in *Q*).

```
(defparameter *Influences*
  '((ExploitsITSystemAsInsider
     (:ImpliedByDisjunction
      (CommitsITExploitation
       (:ImpliedBy (DestroysInformationUnauthorized)
                   (AccessesInformationUnauthorized) ; Ingested: HandlesKeylogger_Event
                   (DisclosesInformationUnauthorized) ; Ingested: CopyFileToWikileaks_Event
                   (StealsInformation))) ; Ingested: CopyFileToCompetitor_Event
      (WarrantsITExploitationSuspicion
       (:ImpliedBy (WarrantsInformationDestructionSuspicion
                    (:IndicatedBy (:Strongly (DeleteFileOnOthersPC_Summary))
                                  (:Moderately (DeleteFileOnLabsPC_Summary))))
                   (WarrantsUnauthorizedInformationAccessSuspicion
                    (:IndicatedBy (:Moderately (AfterHoursLogin_Summary))
                                  (:Weakly (OpenFileOnOthersPC_Summary))))
                   (WarrantsUnauthorizedInformationDisclosureSuspicion
                    (:IndicatedBy (:Strongly (CopyOthersFileToThumb_Summary)
                                             (CopyDecoyToExternal_Summary))
                                  (:Moderately (OpenDecoyFile_Summary)
                                               (AcquireDecoyFile_Summary)
                                               (CopyFileToExternal_Summary))
                                  (:Weakly (CopyFromThumbToOwnPC_Summary)
                                           (CopyOwnFileToThumb_Summary)
                                           (CopyOthersFileToExternal_Summary)))))
       (:RelevantIf (:Locally (:Absolutely (Untrustworthy))))
       (:MitigatedBy (:Locally (:Strongly (HasRole-ITAdmin))))))))
```

Fig. 10. Qualitative specification for probabilistic influences in M_S (with semi-colons prefixing comments in red.)

Position Papers

An Analytic Approach for Discovery

Eric Dull and Steven P. Reinhardt
Cray, Inc.
{edull,spr}@cray.com

***Abstract*—With the widespread awareness of Big Data, mission leaders now expect that the data available to their organizations will be immediately relevant to their missions. However, the continuing onslaught of the ”data tsunami”, with data becoming more diverse, changing nature more quickly, and growing in volume and speed, confounds all but the simplest analysis and the most capable organizations. The core challenge faced by an analyst is to *discover* the most important knowledge in the data. She must overcome potential errors and inaccuracies in the data and explore it, even though she understands it incompletely, guided by her knowledge of the data's domain.**

We have solved customer problems by quickly analyzing numerous dimensions of data to check its sanity and to compare it to expected values. Guided by what we find initially, we quickly move on to further (unanticipated) dimensions of the data; discovery depends on this ability. This approach vitally brings the analyst into direct interaction with the data.

We implement this approach by exploiting the ability of graphs (vertices and edges) to represent highly heterogeneous data. We use RDF as the data representation and SPARQL for queries. We use non-parametric tests, which are readily targeted to any type of data. This graph-analytic approach, using proven techniques with widely diverse data, represents a guidepost for delivering greatly improved analysis on much more complex data.

I. Introduction

Analyst groups are being strongly challenged to understand quickly the insights latent in their organization's data, despite its diversity, changing nature, volume, and speed. We focus on the *discovery* aspect of analysis, where the analyst cannot rely on techniques that have run previously on the given datasets, but rather must explore within the data in a way that cannot be predicted. The organization expects that the analyst will quickly uncover the most important knowledge in the data. Using her knowledge of the data's domain, she must explore the data despite its potential imperfections.

If we define *discovery* as finding a connection in the data of which the analyst was previously unaware, it requires more than just delivering existing capabilities a little faster or more easily used. Rather, it requires enabling a subject-matter expert (SME), *i.e.* the person with the most knowledge and intuition about the data, to explore the data quickly, and even, by appropriate pre-analysis, pulling the SME's eye to aspects of the data that are likely to be most fruitful of further exploration. We have evolved an analytic approach to discovery, implemented in the semantic technologies RDF and SPARQL, that enables rapid progress through analytic questions toward an analytic goal. We have solved high-value customer problems by quickly analyzing numerous dimensions of data to check its sanity and to compare it to our expectations, then moving on to further dimensions of the data guided by what we find initially. Sometimes we analyze data and compare it to our mental expectations. Other times we compare data from one subset (place, time, etc.) to that from another subset, to see if they differ in unexpected ways. Sometimes we analyze the *values* of the data and other times the *connectivity* of the data. Sometimes we create a synthetic grid or discretization, say of geospace and time, to represent data for fast comparison. Discovery depends on being able to compare dimensions quickly, without knowing in advance the dimensions to compare. Bringing the SME into direct interaction with the data is essential to accelerating discovery.

We implement this analytic approach by exploiting the ability of graphs (vertices and edges) to represent the richness of highly heterogeneous data and enable discovery within it. To date, we use RDF as the data representation and SPARQL for queries, queries that can build on each other to focus rapidly on the highest-value knowledge (in the estimation of the subject-matter expert) in the data. (We use *heterogeneous* rather than *semantic*, because this approach is not limited to natural language.) RDF supports complex, dynamic ontologies, though that adds a burden of discovering the current ontology, which we often achieve by summary graphs of vertex-types and the edge-types that connect them. We use Jaccard scoring [6] and non-parametric tests (typically Kolmogorov-Smirnov[7]), which are readily targeted to any type of data when guided by a SME. Other non-parametric tests could easily be used in place of these. RDF and SPARQL are one data format and language that support implementation of this approach, but it may be implemented with other technologies, such as Spark/GraphX from the Berkeley Data Analytic Stack [2].

The graph-analytic approach, using proven techniques with widely diverse data, represents a guidepost for delivering greatly better analysis on much more complex data.

II. Addressing the Core Challenge

The core challenge facing many analytic organizations, illustrated in Figure 1, has at its center a data repository, with both new instances of existing types of data and instances of new types of data flowing into it. Results from existing analytics must flow out to meet existing production needs, while new analytics must be created to discover further knowledge in the data. This paper focuses on this discovery process, based on our experience working with a customer needing to understand traffic and vulnerabilities on its corporate network. In that context, some key entities are IP addresses, ports, protocols, and Internet domain names. Section III.B provides an example of these techniques applied to flow-data

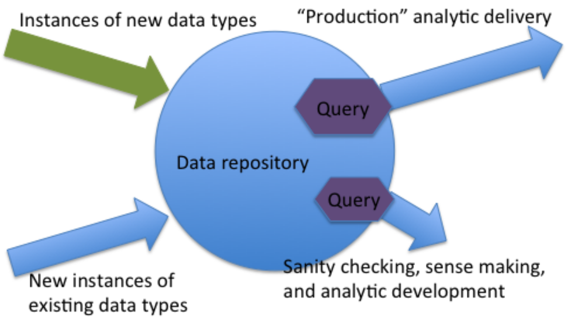

Instances of new data types
"Production" analytic delivery
Query
Data repository
Query
New instances of existing data types
Sanity checking, sense making, and analytic development

Fig. 1. High-level Workflow

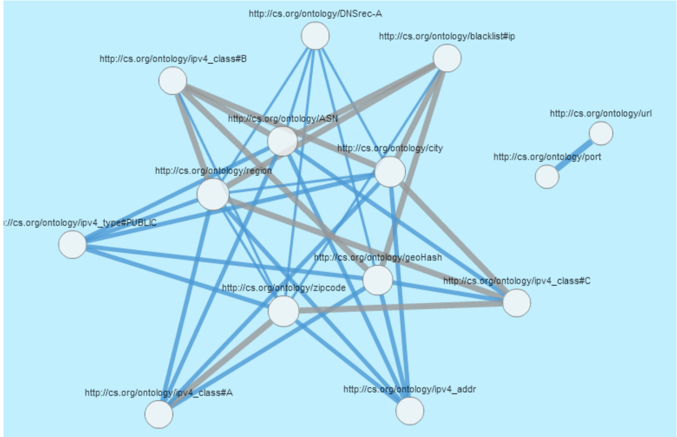

http://cs.org/ontology/DNSrec-A
http://cs.org/ontology/blacklist#ip
http://cs.org/ontology/ipv4_class#B
http://cs.org/ontology/ASN
http://cs.org/ontology/url
http://cs.org/ontology/city
http://cs.org/ontology/port
http://cs.org/ontology/region
://cs.org/ontology/ipv4_type#PUBLIC
http://cs.org/ontology/geoHash
http://cs.org/ontology/ipv4_class#C
http://cs.org/ontology/zipcode
http://cs.org/ontology/ipv4_class#A
http://cs.org/ontology/ipv4_addr

Fig. 2. Discovered Ontology

III. Analytic Approaches for Discovery

We start from the point of view of an analyst with little knowledge about a body of data and proceed to the point of deep knowledge about the data. Many of the techniques we describe are useful at multiple points along this continuum.

Analysis is an iterative process that consists of framing the analytic problem, defining an approach, gathering the applicable data, understanding the biases present in the data (via sanity-checking and sense-making), applying analytic techniques consistent with the approach, determining the analytic results that answer the problem, and documenting the answer to the analytic problem.

A. Sense-making

When an analyst first gets a new corpus of data, even new instances of an existing corpus, the first task is to ascertain the sanity of the data, and then understand it deeply enough to enable further analysis. In practice, an initial sanity-check on data is often necessary, but even then errors in the data may still surface as analysis becomes more precise. For instance, if the data claims to cover the 24 hours of a day, but only has data in the hours 0 through 9, there may have been an error in the software that generates the data. In a heterogeneous context, if flow-data records use IP addresses that are not found in the firewall records, it may be that the data is truly disjoint, or it may be that the IP addresses in the flow data and firewall data have not been constructed the same and hence do not match. In either case, the anomaly needs to be understood before continuing with the data.

With heterogeneous data, sense-making includes discovering the *ontology* (or schema, in relational database terms) in which the data is represented, as that ontology is not explicitly defined anywhere [1]. A *summary graph* depiction of the ontology is shown in Fig. 2, with vertices representing the types of objects in the data. The edges summarize the edges between instances of the types of objects that are the subject and object of triples in the data; edge thickness represents the number of edges between the two types. While this figure is simpler than many real-world ontologies, it illustrates how the structure of the data can be readily represented for to aid an analyst's understanding. Note that the visualization of the ontology as a graph is natural; whether the underlying representation is graph-based or not is immaterial at this level.

A next level of sense-making analyzes the values of a given type of edge. For instance, flow data is often discretized such that a flow cannot last more than an hour. Similarly, TCP/UDP ports are limited to 0-65535. If the data has values outside those ranges, it is suspect. (Additionally, the analyst must ensure that the right units (*e.g.*, English v. metric) are being used.) Not all types of edges will have values that will be readily known by the analyst, but for those that are, this can be a simple way of surfacing the subject-matter knowledge of the expert, by showing the values in the data and letting the analyst respond if the data looks suspect. Beyond correctness, understanding the semantic meaning of a field, *e.g.*, a timestamp, is important. What is the format of the timestamp (absolute, relative)? What is its precision? When was it collected? What specific event is denoted by the timestamp?

Once the basic structure of the data and its values are understood, other questions arise, such as the quantity of the data and whether that indicates that we have all the data we expected or not. *E.g.*, if the data purports to capture all flow data from a 24-hour period, and we know there are about 30,000 flows per second, we should have about 2.6 billion flows. If we do not, we may be missing data, the data may be summarized, or there is some other effect; in any case, the analyst needs to understand the anomaly before proceeding.

Understanding the data values in more detail is another part of sense-making. What are the dominant moments, and, often as importantly, what are the outliers? Analysts know that real-world data is noisy and messy, so will want to avoid actual noise but at the same time want to understand rare but real events accurately. In addition to looking at the total data, we often gain insight by comparing data from entities selected from different ranges in some important dimension, like place or time, to see whether the values in a given dimension also vary; *e.g.*, are the top N most frequent external domains in Internet flow data from the day before yesterday and yesterday

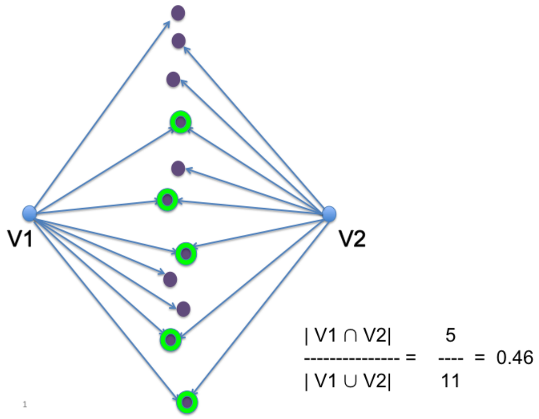


Fig. 3. Example of Jaccard Scoring

similar? Or, is the distribution of bytes per packet the same for systems in Europe as in North America?

This phase of the workflow is critical to an analytic process with meaningful, repeatable results. Performing it requires tools, methods, and (most significantly for time-sensitive analysis) time at the beginning of the analytic process.

B. Techniques

1) Jaccard scoring: Jaccard similarity for comparing two entities is defined as the cardinality of the intersection of values of the two entities divided by the cardinality of the union of their values. An example is shown in Fig. 3, where V1 and V2 each represent the flow data from one hour, with the circles between representing the eight most commonly visited Internet domain names, showing five entities in the intersection from a total of eleven entities in the union. The SME must judge whether that level of variability merits further investigation.

When analyzing Internet flow data, Jaccard scoring can be used to calculate the similarity between two time periods of the top 20 visited Internet domain names. The Jaccard calculation can be done either unweighted or weighted; *e.g.*, for the top-20-domains case, weighting would mean that the similarity between large number of visits to (say) google.com is weighted more than the small number of visits to another domain. Conversely, the weighting can be for rarity; *e.g.*, if we want to know whether two people are similar, the fact that they both visit a domain that is rarely visited population-wide means more than the fact that they both visit a very common domain.

2) Synthetic discretization with Jaccard scoring: The analytic applications discussed above are all examples where the analyst has subject matter expertise. Similarity scoring can also be used when the data is known but the meaning of the data is unknown; *i.e.*, to tell us if two data sets are similar enough that sense making can be skipped, thus semi-automating this critical step at the beginning of the analytic workflow.

Within a semantic context, we can apply Jaccard scoring to the predicate and object types found in a data set. This requires two extensions of the canonical algorithm in section IV.A below. These extensions correct for very popular nodes and handle different types of nodes (categorical similarity).

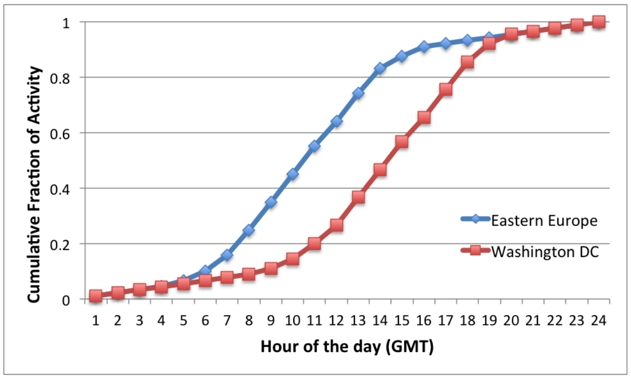


Fig. 4. Example of Kolmogorov-Smirnov Test

3) Kolmogorov-Smirnov: Once similarity scoring demonstrates a high degree of similarity in the predicate- and object-type composition of two data sets, another approach that can be applied is non-parametric goodness-of-fit statistical comparison between degree distributions of subjects with identical semantic types, using tests such as Kolmogorov-Smirnov [7] or Mann-Whitney U. For example, for each semantic type, the degree distribution for each predicate type is generated for the old, analytically-proven dataset and the new, believed-similar dataset. These degree distributions are compared via non-parametric goodness-of-fit statistical tests. The success or failure for each semantic-type/predicate-type pair is noted and then the aggregate success and failure counts are presented to the analyst at the end of the evaluation.

In Fig. 4, we use Kolmogorov-Smirnov to compare two sets of flow data, revealing that the time (measured from GMT) of the activity is distinctly different, with the blue data reflecting the normal work day of eastern Europe and the red data reflecting that of the US East coast.

4) Graph Algorithms: In many cases further insight can be gained from semantic data by directly analyzing it as a graph. For instance, knowing a set of IP addresses and the volume of data between each pair of addresses, the betweenness centrality algorithm will calculate which IP addresses are most *central*, giving insight into which might be most important for covert communication, or most disruptive were they to be disabled. Other graph metrics that may be valuable are the length of the shortest paths between entities, the communities that emerge by analyzing the connectivity of the graph, and the most likely path between entities (vertices). Such analysis of social networks is well known, but any transactional network lends itself to similar analysis. In our work we have used community detection [10], betweenness centrality [3], and BadRank [4].

IV. Implementation with RDF/SPARQL

Much of our work to date has focused on the RDF data representation [9] and SPARQL query language [11] implemented on Cray's Urika-GD^{TM}graph appliance.

A. Jaccard scoring

Both the intersection and union steps in Jaccard scoring map trivially to SPARQL constructs. In the source listing below, an almost-identical subquery is repeated 4 times, once for calculating the cardinality of each set in the intersection and once for calculating the cardinality of each set in the union. The first instance of that subquery is in lines 9-14. It focuses on the first set of data (L11) residing in a named graph `g1`, defined by the `PREFIX` statement (L2), and specifically the data denoted by relationship (edge-type) `:myPred`. Those instances are grouped by the subject, counted within each group, ordered by descending count, and then only the 20 highest-count subjects are retained. Those partial results are joined with the same from the second set by the enclosing query (L7-21), which counts the number of distinct resulting subjects. The other subquery (L22-37) similarly calculates the union, with the sole substantive difference (besides variable names) being the UNION keyword inserted (L30) between the two subsubqueries to denote that results that appear in either subsubquery should be retained. Finally the outer query (L5-38) uses the numerator and denominator values to calculate the final Jaccard score.

```
 1 PREFIX : <http://mydata.org>
 2 PREFIX g1: <urn:g/2014-04-03T10:00:00>
 3 PREFIX g2: <urn:g/2014-04-03T12:00:00>
 4
 5 SELECT (?num/?denom AS ?jaccard)
 6 WHERE {
 7   { SELECT (COUNT(DISTINCT ?s) AS ?num)
 8     WHERE {
 9     { SELECT ?s (COUNT(?s) AS ?s1Ct)
10       WHERE {
11         GRAPH g1: { ?s :myPred ?o1 }
12       } GROUP BY ?s ORDER BY DESC(?s1Ct)
13         LIMIT 20
14     }
15     { SELECT ?s (COUNT(?s) AS ?s2Ct)
16       WHERE {
17         GRAPH g2: { ?s :myPred ?o2 }
18       } GROUP BY ?s ORDER BY DESC(?s2Ct)
19         LIMIT 20
20     }
21   } }
22   { SELECT (COUNT(DISTINCT ?s) AS ?denom)
23     WHERE {
24     { SELECT ?s (COUNT(?s) AS ?s1Ct)
25       WHERE {
26         GRAPH g1: { ?s :myPred ?o1 }
27       } GROUP BY ?s ORDER BY DESC(?s1Ct)
28         LIMIT 20
29     }
30     UNION
31     { SELECT ?s (COUNT(?s) AS ?s2Ct)
32       WHERE {
33         GRAPH g2: { ?s :myPred ?o2 }
34       } GROUP BY ?s ORDER BY DESC(?s2Ct)
35         LIMIT 20
36     }
37   } }
38 }
```

B. Graph Algorithms

Graph algorithms can be executed via SPARQL in two ways. The first is to implement the algorithm in SPARQL, often by a succession of queries [12], with the benefit of having great flexibility to implement the algorithm and the downside of needing to implement the algorithm one's self. The second, for a small set of algorithms on Urika-GD$^{\text{TM}}$, is to call a *built-in graph function* (BGF) in Summer 2014 or later releases [8], with the benefit of ease of execution and optimized performance, for the supported functions. The query below illustrates a simple call to the community-detection function. The INVOKE function calls the designated BGF and the PRODUCING keyword assigns the results to SPARQL variables.

```
CONSTRUCT {
  ?v1 ?count ?v2
} WHERE {
  SELECT ?v1 ?v2 (COUNT(?v2) as ?count)
  WHERE {
    ?v1 :myPred ?v2
  } GROUP BY ?v1 ?v2
} INVOKE yda:community()
  PRODUCING ?v ?communityID
```

V. Conclusion

We have used an analytic approach successfully with several different customers requiring discovery in highly diverse Big Data. The approach grows from initially no knowledge of the data to eventual deep knowledge, by enabling an analyst to interact directly with the data and apply her domain knowledge and intuition. We have implemented this approach with RDF and SPARQL on Cray's Urika-GD$^{\text{TM}}$graph-analytic appliance. We believe this approach represents one way of addressing the critical challenge of quickly discovering deep knowledge in highly diverse Big Data.

References

[1] S. Al-Saffar et al, *Structure Discovery in Large Semantic Graphs using Extant Ontological Scaling and Descriptive Semantics*, Int'l Conferences on Web Intelligence and Intelligent Agent Technology, IEEE, 2011.

[2] M.J. Franklin, *Making Sense of Big Data with the Berkeley Data Analytics Stack*, Proceedings of the 25th International Conference on Scientific and Statistical Database Management. ACM, 2013.

[3] L. C. Freeman, *A set of measures of centrality based on betweenness*, Sociometry, 40, 35-41, 1977.

[4] T.G. Kolda et al. *Generalized badrank with graduated trust*. Technical Report SAND2009-6670, Sandia National Labs, Albuquerque, NM, 2009.

[5] T. Mattson et al. *Standards for graph algorithm primitives*. In High Performance Extreme Computing Conference, 2013 IEEE (pp. 1-2).

[6] P. Jaccard. *Étude comparative de la distribution florale dans une portion des Alpes et des Jura*. Impr. Corbaz, 1901.

[7] F.J. Massey, *The Kolmogorov-Smirnov test for goodness of fit*. Journal of the American Statistical Association, 1951.

[8] D. Mizell et al, *Extending SPARQL with Graph Functions*, High Performance Big Graph Data Management, Analysis, and Mining, IEEE, 2014.

[9] RDF Working Group, *Resource Description Framework (RDF)*, 2004.

[10] E. J. Riedy et al, *Parallel Community Detection for Massive Graphs*, Lecture Notes in Computer Science, Volume 7203, pp 286-296, 2012.

[11] S. Harris et al, *SPARQL 1.1 Query Language (W3C Recommendation 21 March 2013)*, http://www.w3.org/TR/sparql11-query/, 2013.

[12] R. Techentin et al, *Implementing Iterative Algorithms with SPARQL*, Third International Workshop on Querying Graph Structured Data, 2014.

Towards A Topological Framework for Integrating Semantic Information Sources

Cliff Joslyn
Data Sciences and Analytics
Pacific Northwest National Laboratory
Seattle, WA 98103
Email: cliff.joslyn@pnnl.gov

Emilie Hogan
Computational Mathematics
Pacific Northwest National Laboratory
Richland, WA 99352
Email: emilie.hogan@pnnl.gov

Michael Robinson
Mathematics and Statistics
American University
Washington, DC 20016
Email: michaelr@american.edu

***Abstract*—In this position paper we argue for the role that Topological Data Modeling (TDM) principles can play in providing a framework for sensor integration. While used successfully in standard (quantitative) sensors, we are developing this methodology in new directions to make it appropriate specifically for semantic information sources, including keyterms, ontology terms, and other general Boolean, categorical, ordinal, and partially-ordered data types. Given pairwise information source integration principles, TDM can measure overall consistency, and most importantly, reveal cyclic dependencies amongst data sources where conflicts might not be able to be identified. We illustrate the basics of the methodology in an extended use case/example, and discuss path forward.**

I. Introduction and Motivation

There is a need to develop systems to establish situational awareness of events based on multiple real-time information feeds. Information about a typical public event may be available from published news reports, cameras, audio streams, eyewitness blog posts, public twitter feeds, and police information. What are the characteristics of such an information integration problem? What is the significance for modeling choices of the fact that (some of) these input feeds are of a semantic nature? How can we then integrate such multiple feeds to form a holistic operational picture of the relevant situational characteristics, such as participants, identities, attitudes, and preferably content? How do we assess consistency of data values given overlapping measurements (different units, vocabularies, numerical types)? How can we identify locally or globally consistent or inconsistent data, or regions of the network where such conflicts may not be able to be identified?

At present there are no rigorous mathematical techniques deployed to integrate qualitative, semantic information (e.g. from textual analytics) with traditional quantitative signals and sensors data [7]. But there is a provably well-justified mathematical approach to approach this problem. We propose Topological Data Modeling (TDM) techniques from topology, combinatorics, and category theory to address problems in information integration, extending to semantic data sources. The mathematical tools of TDM include machinery like topological spaces, set systems, cell complexes, simplicial complexes, delta complexes, homology and co-homology, and sheafs and co-sheafs to represent both the properties of each analytic, and, most importantly, their pairwise and multiway interactions.

While initially proven to be tremendously valuable in traditional signal analysis (e.g. radar networks or collections of optical cameras [9]), TDM methods using persistent homology, finite topology, and sheaf theory are increasingly penetrating data analytics and knowledge discovery [3]. Extensive theoretical work in sheaf theory [6] leads to powerful detection and inference methodologies in the abstract. When we cast applications into sheaves, the theory does the work of providing a systematic, algorithmic way to globalize data. These methods promise the possibility of critical new capabilities, including:

- Modeling the structural connectivity of information networks, representing multi-way interactions and information overlaps among heterogeneous sensors.
- Modeling the data content flowing within such networks, so that given knowledge of each sensor individually, and knowledge of locally consistent interactions, it can be possible to automatically generate a candidate global view of the integrated sensor network.
- Measuring the overall topology of the resulting network, providing critical information about where cyclic dependencies may hide latent inconsistencies.
- And finally, measuring a network's sensitivity to variation, perturbation, or reliability of the constituent sensors and their connections.

We abstract the concept of a "sensor" from an instrument generating a quantified signal to a generic information process returning a stream of observations, either direct measurements, derived measurements, or the output of an analytic process. We then cast "semantic sensors" more specifically as computational analytics which return *symbolic* information such as keywords, topics, handles, hashtags, proper nouns (individuals, groups, places), and sentiment, including polarity and intensity.

Mathematically, it may be sufficient to distinguish semantic sensors as those whose data types are less than numeric or scalar (integers or real-values, or vectors of these types). This would include Boolean values (polarity), categoricals (keywords), small cardinality ordinals (intensity, sentiment, e.g. high, medium, low), partially ordered entities (ontology nodes), and semantic graph nodes (social network role). Semantic sensor data live in mathematical spaces which are relatively impoverished compared to the richer vector spaces or higher-order structures normally used in TDM. While this has made penetration of topological approaches into data analytics more difficult, more modern methods like sheaf theory, and its use of "categorification", do have the ability to build the needed machinery to support topological representation of these simpler structures. This provides a method to build integrated information networks which combine semantic and quantitative data in a principled way.

More strongly, TDM promises a mathematical approach which is not only sound, but axiomatically necessary, in that theorems indicate that *any* methods for consistency-checking and global modeling of linked sensor networks will recapitulate these TDM methods [1]. TDM promises to support a range of new capabilities such as 1) automatically generating a global model of how sources can be integrated; 2) assessing consistency within the model; 3) measuring the degree of fit of the two models given only partial information about each; and 4) testing for sensitivity with respect to the presence, absence, or credibility of certain sources.

II. Expositional Use Case

Our version of TDM proceeds by specifying some situation in the world about which we have some questions; and where there are many information feeds, "sensors", or "analytics" of different modalities (text, numeric, symbols, ontologies, places) which inform those questions. We require that the user specify only the mathematical form of each input, a mathematical mapping between them pairwise, and which sensors inform which world variables. So while no free lunch, TDM has the ability to handle both quantitative and semantic sources. TDM methods then promise the ability to calculate global and local consistencies. Additionally, and quite importantly, a topological analysis can identify cyclic dependencies amongst information sources, around which it may not be possible to resolve such inconsistencies, requiring intervention or recognition from the modeler.

We now introduce the following true story to drive the example information network. Appropriate for a short position paper, this example was deliberately constructed to be realistic while also illustrating the most important features of our TDM approach. Technical details and example data analysis will await a larger paper in another venue.

On Mayday, 2014, an exuberant group of protesters staged a peaceful demonstration in downtown Seattle in support of immigrant rights and an increased minimum wage. Shortly thereafter, a group of even more exuberant "anticapitalists" meandered through the city streets, from downtown to Capitol Hill, blocking intersections and lighting small fires. Police mostly watched or "escorted" the protesters, but towards the end a half dozen people were arrested, and some tear gas was deployed.[1] While a fine time was being had by all that evening, one of us (Joslyn) was spending a night in in Richland, Washington. There he followed the events of the day through the local KOMO TV news feed and a couple of twitter feeds.

Imagine that in addition to these sources, we additionally had access to overhead video, police scanner audio, Seattle urban transit cams at major intersections, and the feed from the Seattle Times. Fig. 1 shows the overall situation, and how these means might inform our ability to track a collection of "state variables":

$S =$ **Size of the crowd:** An integer.

$O =$ **tOpic being protested:** Terms like "immigrant rights", "minimum wage", or "anti big business" are normalized into an ontology, each being a node in a partially-ordered semantic class hierarchy.

$P =$ **Place:** A categorical variable like "1st and Pine" or "Broadway".

$I =$ **Intensity:** An ordinal variable: "high", "medium", "low".

$L =$ **vioLence:** A Boolean variable: "present" or "absent".

$R =$ **Role:** Another categorical variable, reflecting the kind of person present, for example "protester", "police", "bystander", or "press".

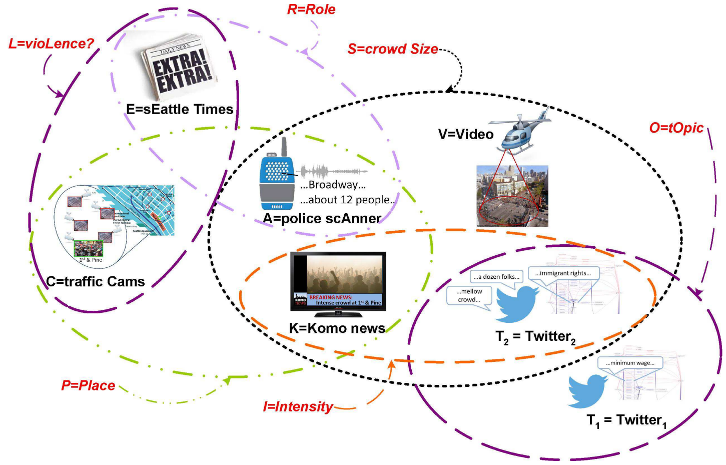


Fig. 1. An information integration scenario: multiple sensors partially informing multiple state variables.

We can cast each information source as a separate sensor or analytic, with structure as follows:

A= **police scAnner:** A speech recognizer has been trained to extract specific information about crowd size and location from speech like "I see about 12 people here at 1st and Pine, 4 police and 8 protesters".

C= **transit Cameras:** Cameras at specific intersections can show when the crowd has reached those locations, and whether violence is present.

E= **sEattle Times:** An anlaytic deployed against the local newspaper web feed to parse out information about the presence of people in certain roles and the presence or absence of violence.

$K =$ **KOMO News:** The news broadcast shows a video feed of crowds with a chyron showing the specific intersections, and video analytics are trained to estimate crowd sizes and intensity.

$T_1 =$ **Twitter$_1$:** A text analytic extracts keywords to identify protest topics.

$T_2 =$ **Twitter$_2$:** A different text analytic extracts keywords to estimate topic, crowd size, and intensity.

$V =$ **overhead Video:** An algorithm is used to estimate the number of people shown in a live video stream.

We model the sensors and their overlapping coverage by letting $X = \{P, S, O, I, L, R\}$ be the set of **state variables** and $Y = \{A, C, E, K, T_1, T_2, V\}$ be the set of **sensors**. Then table I shows the relationships between these sensors and the state variables they inform. We cast Table I as a binary relation $B \subseteq X \times Y$. Then Fig. 2 shows B as a set system (undirected hypergraph) $B(X) \subseteq 2^Y$ on the sensors Y. The variables $x \in X$ (i.e., the columns of B) are represented (in red) as subsets $B(x) \subseteq Y$ of the sensors (in black) which inform them.

[1] http://www.huffingtonpost.com/2014/05/02/seattle-may-day_n_5253707.html

	S	O	P	I	L	R
	crowd Size	tOpic	Place	Intensity	vioLence?	Role
	Number	Ontology term	Intersection	Level	T/F	Police, Protester
	Scalar	Partial Order	Categorical	Ordinal	Boolean	Categorical
A =Police scanner	√		√			√
C=Transit cams			√		√	
E=sEattle times					√	√
K=KOMO News	√		√	√		
T_1 =Twitter1		√				
T_2 =Twitter2	√	√		√		
V =Overhead video	√					

TABLE I. SENSOR STRUCTURE.

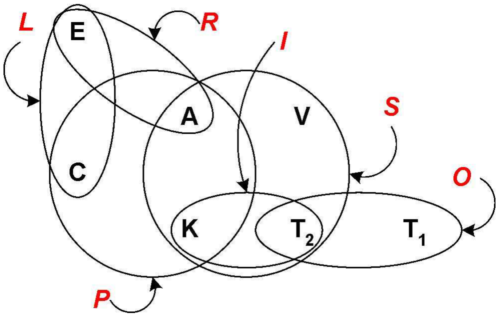


Fig. 2. Representation of sensor structure B as a set system.

Fig. 3 shows $B(X)$ as a combinatorial structure called an "abstract simplicial family" [4] with simplices $B(x), x \in X$ of dimension $|B(x)|-1$. Note that $B(S)$ is the (solid) tetrahedron $\{A, K, V, T_2\}$, with the $\{A, T_2\}$ edge underneath, indicating the four-way interaction of the sensors through the variable S. Similarly, $B(P)$ is the filled-in triangle $\{A, C, K\}$, while the triangle $\{A, C, E\}$ is *not* filled in, rather consisting of the three distinct edges $\{A, C\}$ (for P), $\{C, E\}$ (for L), and $\{E, A\}$ (for R). Also, the edges $\{K, T_2\}$ (column $B(I)$) and $\{T_1, T_2\}$ (the column $B(O)$) are called out from the table and shown in blue, as are the edges $\{C, E\}$ and $\{E, A\}$ (but not $\{A, C\}$).

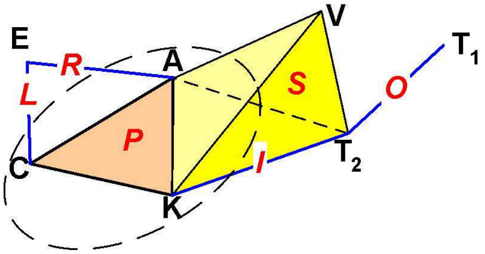


Fig. 3. Representation of B as a simplicial family. The dashed ellipse is exploded in Fig. 4.

While $B(I) \subseteq B(S)$, none of the other faces are pairwise inclusive, and so they comprise the maximal faces of an abstract simplicial complex (ASC), which further contains all the included sub-faces (all triangles, edges, and vertices). The characteristic f-polynomial $x^3 + 5x^2 + 11x + 7$ of the ASC serves to enumerate the counts and dimensionalities of all 24 faces present, not just the "listed" ones read off the table: one (3D) tetrahedron, five (2D) triangles, eleven (1D) edges, and seven (0D) vertices. Abstraction to an ASC allows easy tracking of all k-way interactions dually amongst sensors and variales. Topological features of the connectivity pattern can be identified, including loops, voids, etc., where potential informational feedbacks can result in faulty conclusions. In our case, the ACK triangle can establish consistency around place P, while the ACE loop may yield assignments which are impossible to resolve consistently amongst all three sensors.

We can represent the 24 faces (interactions) distinctly, but for brevity, we only show the 7 associated with the variable P in a "sheaf" diagram in Fig. 4. Here each node shows some combination of the sensors A, C, and K above (black), and the corresponding variables they inform below (red).

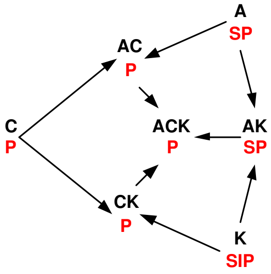


Fig. 4. Sheaf representation of the sensors A, C, K informing the variable P. The arrows are functions transforming data on faces into a common form.

The sheaf in Fig. 4 shows not just all the combinations of sensors, but how they can be mapped into each other to measure consistency. Continuing our drilldown, Fig. 5 shows this in detail for just the $A \mapsto AK$ edge of the sheaf, showing the sensor A (the police scanner) in interaction with the sensor K. Since sensor A reads off in crowd size, role, and location, this is in the form of a three-way data tensor as shown on the right. A and K share only S and P in common, so the matrix projects over R and aggregates S into the two-way tensor shown on the left, reported up from K. Here we can see that there is a match at City Hall (20 total people); a match for Main Street (5 police and 5 dozen bystanders yields 65 total people, which is in the interval $[50, 100]$); but finally a failure at Broadway ($26 \not\in [5, 10]$).

Note how the semantic information is encoded in the various linear objects. Through the process of "categorification", the semantic variables of role and place (both categoricals) have unique positional assignments, as reflected in the block structure of the central matrix, called the "restriction map". This kind of categorification supports the integration of quantitiative data with the mathematically weaker data types typically used for semantic information.

An assignment of data to the sensors which yields consistency over some of the faces is called a "local section" over

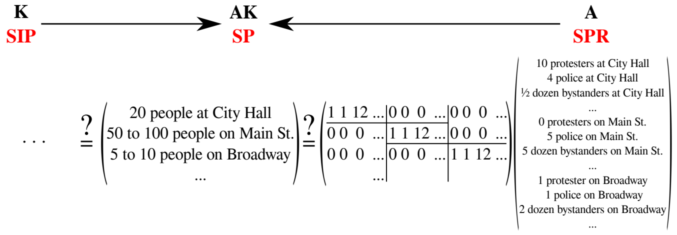


Fig. 5. One edge of the sheaf in detail, showing the mapping of the sensor A against K, checking for consistency.

those faces. Fig. 6 shows a local section over the AC edge and the isolated vertex K, but no data linking AC to K, which is just reporting the weather. Fig. 7 shows a global section over the whole P triangle, indicating agreement of all Place sensors. Both the degree of *consistency* and the degree of *completeness* can thus be measured over this whole portion of the sheaf.

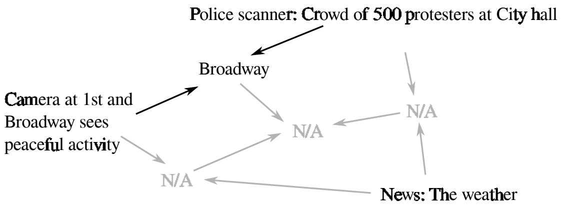

Police scanner: Crowd of 500 protesters at City hall
Broadway
N/A
N/A
N/A
Camera at 1st and Broadway sees peaceful activity
News: The weather

Fig. 6. A local section only over AC and K.

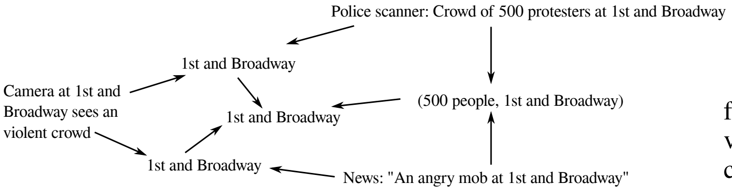

Police scanner: Crowd of 500 protesters at 1st and Broadway
1st and Broadway
Camera at 1st and Broadway sees an violent crowd
1st and Broadway
(500 people, 1st and Broadway)
1st and Broadway
News: "An angry mob at 1st and Broadway"

Fig. 7. A global section over all sensors informing P.

Can this approach be expanded across the entire ASC to measure consistency globally? The A, C, E triangle has no three-way interaction, only the three pairwise interactions. Thus fixing data at one vertex (say A), can constrain another (say C), which in turn can constrain E, but there is no way to guaurantee that there will then be final consistency needed between E and A. Knowing this loop is present is critical to the modeler, and identifying its presence (and those of more complexity) involves calculating the homology of the ASC, or the co-homology of the sheaf. Once identified, the modeler can be informed of the risk, and allowed to mitigate or address it.

III. Path Forward

This small position paper is intended to evoke the spirit and flavor of our TDM approach to semantic information integration. The path forward to a more complete expression of this idea obviously begins with encoding realistic datasets to demonstrate operaion of actual algorithms in our example.

We are showing Boolean methods for local sections: quantities and qualities either match exactly, or satisfy some crisp condition like $65 \in [50, 100], 26 \not\in [5, 10]$. We are also exploring a mathematical theory of "approximate sections", which could provide more robust inferences in the presence of uncertainty. We will establish distances between numerical and non-numerical quantities, which can be aggregated to provide a quantitative degree of match. We will then additionally require the user specify distances measures between data types in addition to their types. In the case of fully semantic data, like the keyterms of ontologies, we could use order metrics on class hierarchies [8], which we have previously established in the context of ontology management [5].

Where sheaves provide a bottom-up view of integrating existing sensors covering certain variables, their dual "co-sheaves" (where the arrows of Fig. 4 are reversed) support "world models" which can specify the structure of sensors *needed* to cover variables of interest (see Fig. 8). Linear duality between sheaves and cosheaves corresponds to the duality between sensor-centric and world-centric modeling disciplines. Recent results on "sheaf and co-sheaf duality" [2] allow construction of explicit joint world/source models, so given a partial world model and a partial source model we may measure degree of fit and seek sensitivity analysis to source variations using "topological persistence".

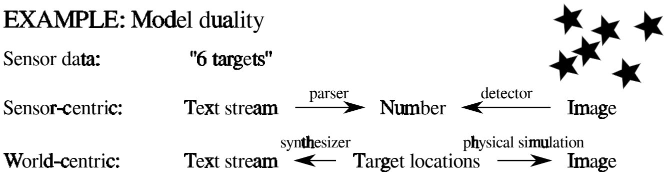


Fig. 8. A sensor-centric sheaf model of a text-image integration, together with its dual world-centric model as a co-sheaf.

IV. Acknowledgements

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions, and/or findings contained in this article/presentation are those of the author(s)/presenter(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Distribution Statement "A" (Approved for Public Release, Distribution Unlimited). This is PNNL-SA-105411.

References

[1] G. Bredon. *Sheaf theory*. Springer, 1997.

[2] Justin M Curry. Sheaves, cosheaves, and applications, 2014. http://arxiv.org/abs/1303.3255

[3] Robert Ghrist. Barcodes: The persistent topology of data. *Bulletin of the American Mathematical Society*, 45:1:61–75, 2007.

[4] Jeffrey Johnson. *Hypernetworks in the Science of Complex Systems*. Imperial College press, 2013.

[5] Cliff Joslyn, Patrick Paulson, and Amanda White. Measuring the structural preservation of semantic hierarchy alignments. In *Proc. 4th Int. Wshop. on Ontology Matching (OM-2009), CEUR*, volume 551, 2009.

[6] M. Kashiwara and P. Schapira. *Categories and sheaves*. Springer, 2006.

[7] B. Khaleghi, A. Khamis, and F. Karray. *Multisensor data fusion: a review of the state-of-the-art*. Information Fusion, 2011.

[8] Chris Orum and Cliff A Joslyn. Valuations and metrics on partially ordered sets, 2009. http://arxiv.org/abs/0903.2679v1

[9] M Robinson. *Topological Signal Processing*. Springer-Verlag, 2014.