Dynamic Data Relevance Estimation by Exploring Models (D^2REEM)

H. Van Dyke Parunak
Soar Technology, Inc.
3600 Green Court, Suite 600
Ann Arbor, MI 48105 USA
van.parunak@soartech.com

***Abstract*—Analysts in many areas of national security face a massive (high volume), dynamically changing (high velocity) flood of possibly relevant information. Identifying reasonable suspects confronts a tension between data that is too atomic to be diagnostic and knowledge that is too complex to guide search. D^2REEM (Dynamic Data Relevance Estimation by Exploring Models) is a knowledge-based metaheuristic that uses stochastic search of a graph-based semantic model to guide successive queries of high-volume, high-velocity data. We motivate D^2REEM by considering the nature of knowledge-based search in high-volume, high-velocity data and reviewing current tools. We then outline the D^2REEM metaheuristic and describe the state of progress in applying it to a range of model types, including geospatial movement, behavioral models, discourse models, narrative generators, and social networks. Finally, we outline work that needs to be done to advance the D^2REEM agenda.**

Keywords*—*retrieval, querying, semantic models, big data, stochastic search, any-time methods

I. Introduction

Analysts in many areas of national security face a massive, dynamically changing flood of possibly relevant information. "Big data" is typically described in terms of Volume (the amount of data), Velocity (how fast it changes), and Variety (the diversity of data formats); our concern here focuses on high-volume, high-velocity data. Activities of crucial interest can be expected to leave many "footprints" in available data, but identifying reasonable suspects confronts a tension between *data that is too atomic* to be diagnostic and *knowledge that is too complex* to guide search.

The *data* problem is that no single data item is diagnostic of an attack. Any one data item that might be part of an attack could also be part of a benign scenario. For example, a purchase of fermentation equipment might be a precursor to anthrax cultivation...or to opening a microbrewery. A new dissertation on gene splicing in microbes might point to a potential perpetrator...or just a promising new assistant professor. In data retrieval terms, static single-item queries give very low precision in identifying the overall event.

The *knowledge* problem is that while we can capture overall patterns of behavior that are diagnostic, matching them against massive data is combinatorially prohibitive. Representations that are available include discourse models that capture the different forms a conversation in social media might take [1, 2], hierarchical task networks (HTN) that capture goal-oriented behaviors [3, 4], social networks that show possible connections and flows among people and organizations [5, 6], and narrative models that capture causal dependencies [7]. Such a structure covers many possible behaviors, depending on which combinations of constraints are satisfied. If we could match such a structure against data, we would expect very high precision and recall. However, realistic structures can grow very large (for instance, an HTN might contain hundreds or thousands of atomic behaviors and constraints), and naïvely matching such a structure against massive data all at once is combinatorially prohibitive.

This paper describes D^2REEM (Dynamic Data Relevance Estimation by Exploring Models), a knowledge-based metaheuristic that uses stochastic search of a semantic model to guide successive queries of high-volume, high-velocity data. Section II explores the challenge that D^2REEM addresses and the current state of the art. Section III outlines the D^2REEM metaheuristic. The heart of D^2REEM is a knowledge-based model of the domain, and Section IV reviews several classes of models to which D^2REEM may be applied and documents our success so far. D^2REEM is a work in progress. Section V identifies a series of next steps for advancing this approach to semantic-driven search of big data. Section VI concludes.

II. The Challenge and Prior Work

Figure 1 summarizes the challenge that D^2REEM addresses. Static single-record queries are simple, but can be efficiently applied to high-volume high-velocity data. Conventional matching methods are too inefficient to apply knowledge-rich patterns to such data. D^2REEM is a novel way to match complex patterns to big data.

For years, the staple of information retrieval has been the record-oriented query, in which the analyst describes single data items that might be of interest. Static queries can be matched very efficiently, but their relevance depends on the state of knowledge about the world, which changes with each new piece of information.

The last 25 years have produced an explosion in graph databases, that is, databases that capture semantic relationships among data items in a graph structure. Graph databases can be

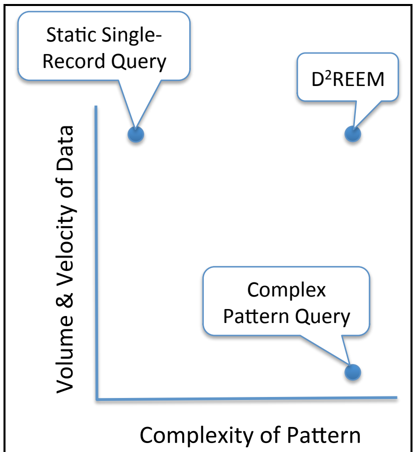

Figure 1: D²REEM 's Advantage

used to answer a range of queries in such data, including subgraph matching (does a specified pattern appear), shortest path discovery, path comparison, and computation of aggregate graph properties. Our focus in this paper is on subgraph matching. Queries against graph databases are done by specifying constraints over multiple nodes, such as a subgraph of the database, or a path that satisfies certain criteria, or specified aggregate characteristics of the graph [8]. Examples of such query languages are Cypher for Neo4J [9], or XPath for XML [10], or SPARQL [11] for RDF [12].

Graphs are a natural way to capture a knowledge model, but classical graphical query languages have several disadvantages for knowledge-based subgraph matching.

- They are generic to any graph-structured data, and do not take advantage of specific semantics in various kinds of graphical models. We wish to exploit the knowledge in a model.
- They require the entire query to match a subset of the data. We would like to search the data against a graphical structure (such as a hierarchical task network [HTN]) that expresses a range of possibilities, and identify coherent subsets of the pattern that the data support.
- In general, graph matching is intractable [13], with either exponential or NP-complete complexity in the size of the query. Thus queries must be kept small [8]. We wish to exploit large knowledge models.
- Graph databases require the data to be represented as a graph. We address high-volume high-velocity data streams (such as social media) where such preprocessing is not feasible.

III. The D²REEM Metaheuristic

D²REEM is a metaheuristic, a high-level procedure that guides a lower-level process (in this case, record-level querying). Like many metaheuristics (e.g., genetic algorithms, ant-colony optimization, swarm optmization, artificial immune systems), its methods are strongly inspired by biological models.

TABLE I. Comparison of D²REEM with Subgraph matching in Graph DBs

	Graph DB	D²REEM
Query Size	Small Expresses complete structure of interest Search is for the entire query graph	Large Describes a range of structures of interest Search is for a matching subset
Data	Graph-structured	Record-structured
Query Semantics	Implicit Depends on use of same graph grammar for query and data	Explicit Enforced by PSE and EPM
Matching	Match entire query graph against data	Repeatedly match most relevant query node against data
Processing	Focus is on matching query graph against data graph Complexity is NP complete (subgraph isomorphism)	Focus is on exploring query graph in light of current data, and pursuing information on most relevant node Complexity is linear in size of knowledge model

In this section we introduce the metaheuristic, then explore two of its key components in more detail. The next section discusses classes of knowledge models to which it can be applied, and surveys our experience so far with each of them.

A. *Overview*

D²REEM shifts the focus of computation in doing knowledge-based exploration of big data. It moves computation away from matching the model against the data, and toward executing a process over the model that embodies the distinctive semantics of the model. Table I summarizes the differences between D²REEM and subgraph matching in a graph database.

Because D²REEM works with data as a stream of records, rather than a pre-processed graph, it must issue many record-level queries in order to match a knowledge model. It does this by executing a continuous cycle (Figure 2). Repeatedly, D²REEM

- explores the current state of the model,
- updates the priority for learning more about each node in the model,
- adaptively generates a query for the highest-priority node, and
- updates the model with what is learned from that query.

The queries can be posed to any data source, and do not

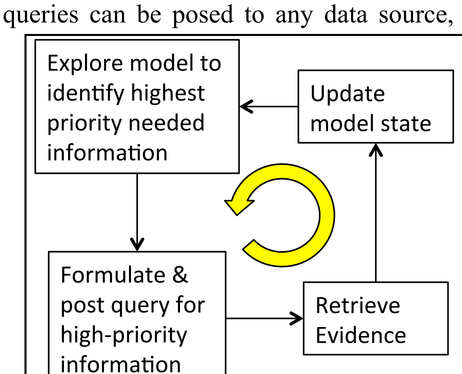

Figure 2: Basic D²REEM Processing Cycle

require predefining relationships among separate data items. The relationships among retrieved nodes are computed by exploring the model, not by a complex matching process, a strategy similar to graph simulation [14] (though unlike that work, we do not require that the data already form a graph).

Figure 1 shows the result. Static single-record queries can be applied to big data, but cannot capture complex patterns among records. Graphical databases can express patterns, but computational complexity forces the patterns to be smaller than a realistic behavioral model, and the data must be small enough and stationary enough to preprocess into a graph. By taking advantage of model semantics, D^{2}REEM can match very large knowledge-rich patterns (with thousands of nodes) against high-volume, high-velocity data streams that are not in graphical form.

Figure 3 shows the D^{2}REEM architecture. The heart of D^{2}REEM is a Graphical Knowledge Model (GKM) with two characteristics:

- Edges in the graph represent causal or other sequential dependencies between nodes, so that a trajectory is a possible evolution of the world, and
- The likelihood of visiting a given node can be modulated by evidence attached to the node.

The Polyagent Sampling Engine (PSE) continuously samples alternative trajectories through the GKM to generate a distribution over possible trajectories reflecting current knowledge of the domain. The Evidence Prioritizer and Marshaller (EPM) examines these distributions to identify nodes about which more information would be useful, issues queries to retrieve that information, and updates the GKM with the results. The PSE's ongoing exploration takes account of this new information, modifying the distributions over trajectories, and thus leading to new rounds of queries, implementing the processing cycle shown in Figure 2.

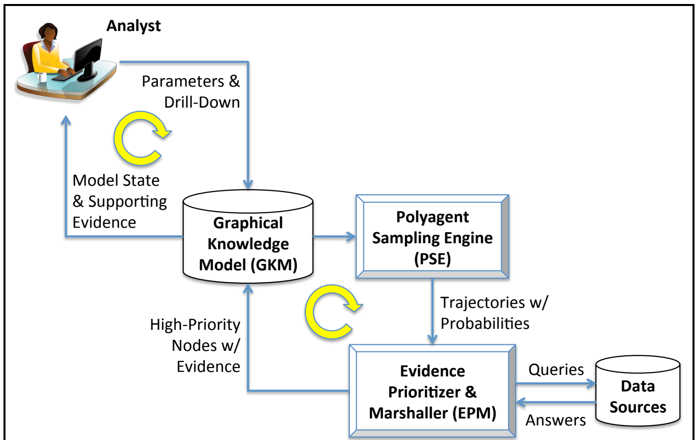

Figure 3: D^{2}REEM Architecture.

B. Polyagent Sampling Engine

By construction, each trajectory through a GKM corresponds to a possible instance of the dynamics implicit in the graph. Evidence currently on each node of the graph modulates the probability assigned to trajectories involving that node. We wish to construct a distribution over all possible trajectories. An approach we have found particularly tractable over many types of GKM is polyagent sampling.

A polyagent is a set of agents that collectively explore possible trajectories for a single entity or behavioral instance of interest. It consists of a single persistent coordinating agent (the "avatar"), which continuously generates a stream of simple agents ("ghosts"), each exploring a single trajectory. Figure 4 shows a polyagent sampling possible paths through a geospatial domain.

Ghosts have three biologically-inspired characteristics: they are manifold, apoptotic, and tropistic. "Manifold" means that many of them explore the domain in parallel, like multiple ants in an ant, or multiple chromosomes in genetic evolution, or multiple antibodies in an immune system, or agents in swarm optimization. "Apoptotic" means that they die after a fixed number of cycles. Thus the avatar can continue to generate new ghosts without overloading the system. "Tropistic" means that they move based on the characteristics of their environment, like ants. Physical ants plan paths through complex environments by depositing and responding to chemical pheromones. Polyagent ghosts respond to "digital pheromones," scalar fields maintained on the nodes of the GKM. These fields may reflect evidence supporting or refuting a given node. In addition, ghosts deposit a presence pheromone on each node that they visit. The normalized presence pheromone over the entire graph gives a probability distribution over possible trajectories of the entity that the polyagent represents.

While the immediate inspiration of the PSE is biological, its mathematical underpinnings are based on Monte-Carlo tree search (MCTS) [15, 16], which explores multiple descendants of a single node to estimate the probability with which that node should be visited. In MCTS, the graph being explored is a game tree, in which the same game rules are applied in expanding each node. The PSE generalizes this concept to other graph structures, taking advantage of their distinctive semantics in the decision rules used by the ghosts and the digital pheromone fields they manipulate.

C. Evidence Prioritizer and Marshaller

The EPM has three functions:

- Based on the distribution of trajectories through the GKM determined by the PSE, identify the nodes for which additional information would be most valuable.
- Formulate and execute queries that will provide more information on the selected nodes

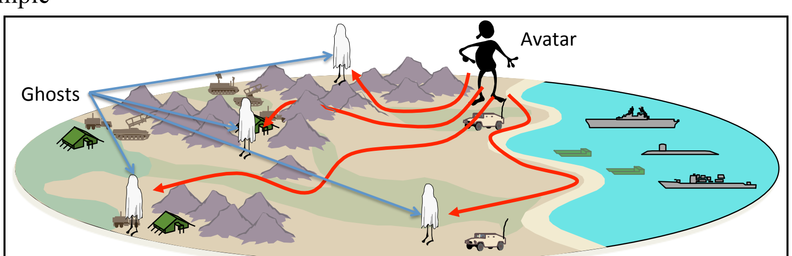

Figure 4: A polyagent (one avatar and four ghosts) in a geospatial domain

- Update evidence on the selected nodes based on the results of the queries.

We consider these in turn.

Identify nodes to guide queries.—Intuitively, D^{2}REEM estimates the relevance of candidate queries based on the nodes for which additional information would be most valuable. The precise sense of "valuable" depends on the kind of GKM that is guiding the search, and the decisions that it is guiding. Here are some alternatives that are useful in different settings. In the next section, we give further examples of each of these.

In sparse environments, the most valuable query is one *most likely to yield a hit*. In our PROPS system, polyagent sampling over a geospatial lattice generates candidate trajectories for adversaries, and the most probable trajectory guides the decision of where to deploy scarce surveillance assets to increase the probability of detecting an adversary.

One might try to *maximize some global measure* over the GKM. One use for an HTN in D^{2}REEM is to model a potential adversary's behavior (e.g., mounting a biological attack). In an HTN (using the rTÆMS dialect [4]), each leaf task that is executed contributes to the quality that accumulates at the root, and the higher that quality, the better the objective is achieved. By examining a set of possible trajectories identified by the PSE, the EPM can identify which trajectory would yield the highest root quality. If the HTN models adversarial behavior, this trajectory is most consistent with the adversarial intent we are seeking to detect. In this case, we want to select the nodes for which gaining more information might increase the probability of that trajectory.

In some cases, the nodes about which we want to learn more are those for which more information would *sharpen the distribution over alternative trajectories*. We estimate the effect of this choice by changing the evidence levels for various nodes in copies of the GKM and run the PSE on them, then compare the resulting distributions.

Formulate and Execute Queries.—The EPM submits queries to external data sources for those nodes that have been identified as of highest priority. Currently, we construct queries for each node manually in the course of formulating the GKM, and the EPM retrieves the specified query and submits it.

Update Node Evidence.—The EPM updates the evidence supporting the node on the basis of the response to the query. This change modulates the ongoing execution of the PSE, potentially changing the highest priority nodes in the next invocation of the EPM and directing the search process to the most relevant potential data.

IV. EXAMPLES OF D^{2}REEM MODELS

The heart of D^{2}REEM is a semantic model of some facet of the real world. We have identified numerous such models, and demonstrated various facets of D^{2}REEM on them. This section outlines these examples. For each, it discusses

- how the model supports the two requirements identified in Section III.A (trajectories represent possible evolutions of the world; evidence on nodes modulates probability of trajectory)
- how it supports the PSE and EPM (in particular, what makes a node "relevant"), and
- what aspects of D^{2}REEM have been implemented in it.

A. Movement on Geospatial Maps

The most mature class of GKM for polyagent sampling is the geospatial lattice, whose nodes correspond to tiles of the environment and whose edges represent adjacency among tiles [17].

A trajectory represents the movement of a target, and the probability that a trajectory visits a node depends on externally-provided information such as terrain, presence of friendly or adversarial forces, and combat activity. The cumulative distribution of presence pheromone thus reflects the probability of encountering the target as a function of location.

In the DARPA RAID project, we applied the PSE on such a model to urban combat. Figure 5 shows the close correlation of predictions of red force movement in a human-commanded wargame, compared with the actual movement of troops. Quantitatively, the PSE produced more accurate forecasts than both experienced human observers and game-theoretic or Bayesian reasoners [18].

In the ARL PROPS project, we used the PSE on a geospatial lattice to direct collection management. The relevance criterion in this case is to give priority to queries (intelligence requests) on areas most likely to generate a hit.

PROPS is the most mature implementation of the D^{2}REEM metaheuristic to date, including ongoing PSE exploration of the knowledge model, dynamic query formulation, and updating of the knowledge model.

B. Hierarchical Task Networks

Goal-oriented behavior by intelligent agents is often represented with a hierarchical task network (HTN) [4, 19]. Figure 6 is a fragment of an HTN model for a mix of benign and nefarious cyber-activities. The nodes are actions, and are joined by two kinds of edges: subtask edges (solid) that connect a higher-level task (the goal) to lower-level tasks that

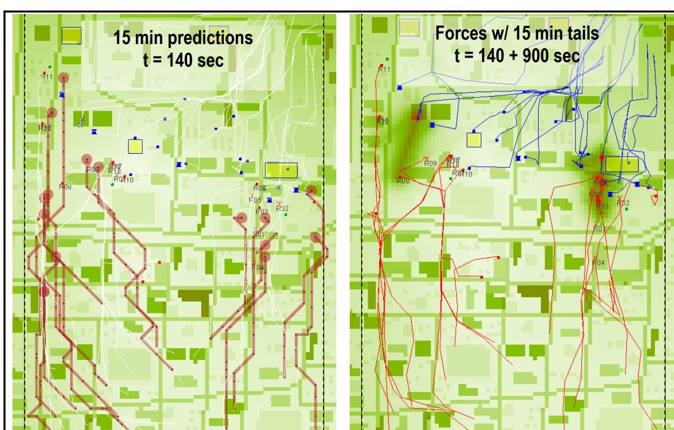

Figure 5: Example geospatial forecasts

carry it out, and sequence edges (dashed) that reflect precedence constraints. These precedence constraints are inherited by the leaves of the HTN. The graphical language in the figure is a simplification; our full formalism, derived from the TÆMS language [19], is much more sophisticated. In TÆMS, successful execution of a task generates "quality," a scalar value, that propagates upward via combination rules. The degree to which a sequence of leaf actions satisfies a top-level goal is measured by the amount of quality that accumulates at that top node.

Polyagent sampling explores alternative trajectories through the leaf tasks. Each trajectory reflects a sequence of actions that an agent might execute in the world. The probability that an agent's next step will move to a given task depends on the task's feasibility (the satisfaction of its prerequisites), its desirability (based on the degree to which higher-level tasks have been achieved), and evidence for the task from the external world (provided by the EPM).

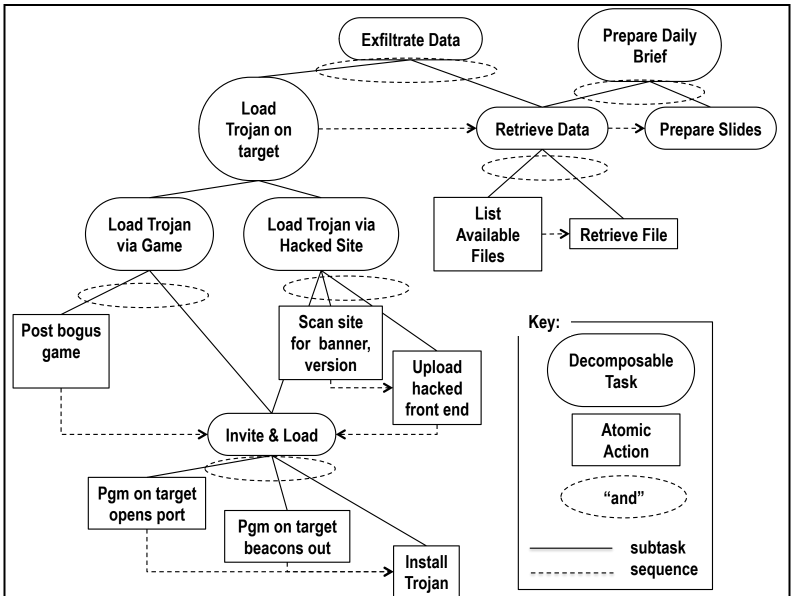

Figure 6: HTNs can model the structure of complex activities.

The HTN is an example of a GKM for which the value of generating a query for a node depends on a global characteristic of the model, namely, the change in the quality of the root node that a response to the query might generate. We have demonstrated the PSE on HTNs [4, 20], but not yet implemented an EPM for it.

C. Social Networks

We represent a social network [5] as a bipartite graph, in which one set of nodes represents people or organizations, and another indicates class of transaction. Several different kinds of transaction are possible, including communication, transfer of wealth, transfer of power (e.g., by confrontation), or transfer of status (e.g., by endorsement). Figure 7 is an example social network in our PSTK system (Power Structure ToolKit), in which the Agents are people and the bar graphs between them represent the levels of the different transaction types (in this example, Political, Military, Economic, Social, from the PMESII ontology).

A trajectory in a social network indicates a sequential transfer of social capital. For example, one may seek evidence for a money laundering operation that moves a financial payment makes its way through a series of organizations. Evidence of a specific transaction increases the likelihood of a transition from one agent to another.

Our current PSTK system explores possible flows using specialized processes residing on each agent, not the PSE. We have not implemented a EPM for social networks. If one is seeking to identify sequences of transactions, the relevance of a node to generate a query is measured by the degree to which additional information on that node would sharpen the distribution over alternative trajectories. For example, a node that is shared by several emerging trajectories would not rank as high as one that is unique to a single candidate trajectory.

D. Narrative Space Models

A Narrative Space Model (NSM) captures a set of many possible narratives that could explain the evolution of a situation [7, 21], and is an external representation for the mental activity of an analyst who is seeking to explain how a given state of affairs might come about. Each node in an NSM consists of a statement about the world to which belief may be assigned. The NSM has an edge from one node to another just if the first statement followed by the second forms a coherent segment of narrative.

Each trajectory through an NSM represents a coherent narrative about how the world might evolve from the origin to the destination. Figure 8 shows an abbreviated NSM that captures ways that al-Assad might stay in power or lose power in Syria. The '????' notation on edges between nodes are placeholders for edge weights that the PSE fits based on evidence on the nodes. In an NSM, external evidence for (against) an individual node increases (decreases) the probability of trajectories including that node.

We have implemented the PSE on NSMs, and modulated its behavior by attaching external evidence to nodes in the

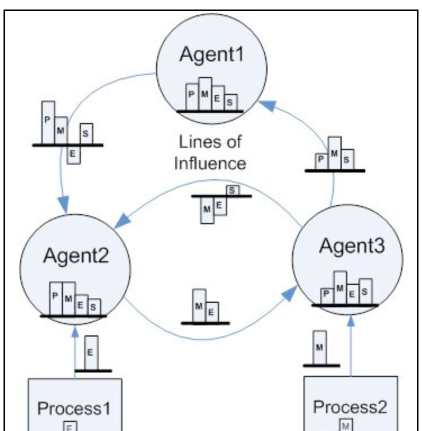

Figure 7: Social network

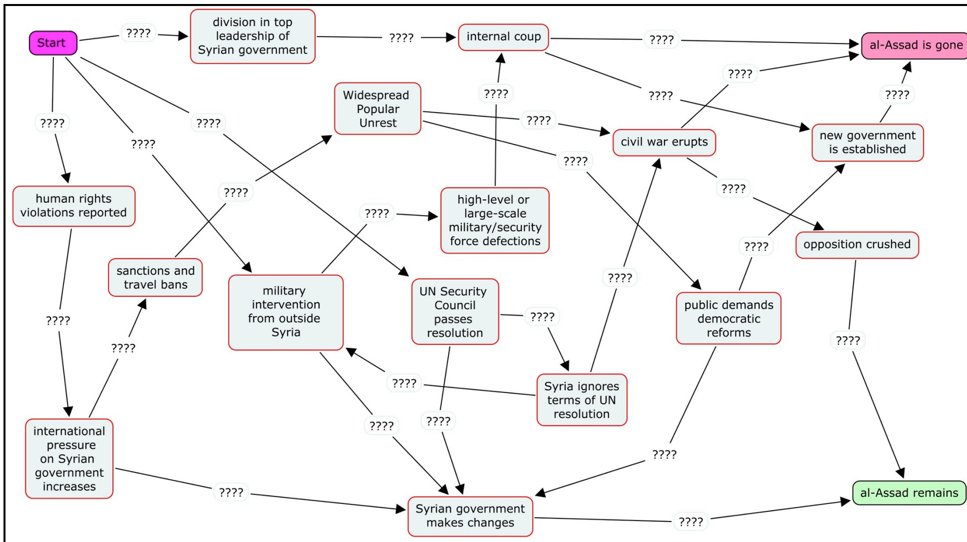

Figure 8: Abbreviated Narrative Space Model for Syria

NSM. In our work so far, this evidence has been attached by a human analyst, not by the EPM. Since our interest is in identifying the most likely narrative given the evidence available to date, the EPM for a NSM would weight nodes based on how much evidence for a given node would sharpen the distribution over emerging narratives.

E. Discourse Models

Dooley Graphs [2] reflect a speech-act view of discourse [22], in which each utterance seeks to accomplish something (e.g., Solicit an action or a statement, Inform, Commit, or Refuse). In a coherent conversation (a sequence of speech acts), later utterances may be related in different ways to earlier ones: they may Respond, Reply, Resolve, or Complete them. Detecting such coherent conversations from a high-volume, high-velocity stream of data (for example, a Twitter feed) would make a great contribution to surveillance activities.

There are a number of ways one could graph a sequence of utterances, depending on what one chooses as the nodes.

- The nodes could represent specific *utterances*, and edges would reflect the sequences among them. This representation loses critical information about who issues each utterance.
- One might analyze the conversation to characterize different states that it could assume, and then represent each *state* as a node, with edges representing possible state transitions. A state model, like an utterance model, deemphasizes the participants, and in addition makes it difficult to distinguish specific conversations.
- We could represent *participants* as nodes, with edges representing utterances from the source to the target. Like the state model, the participant model does not clearly capture the progress of an individual conversation.

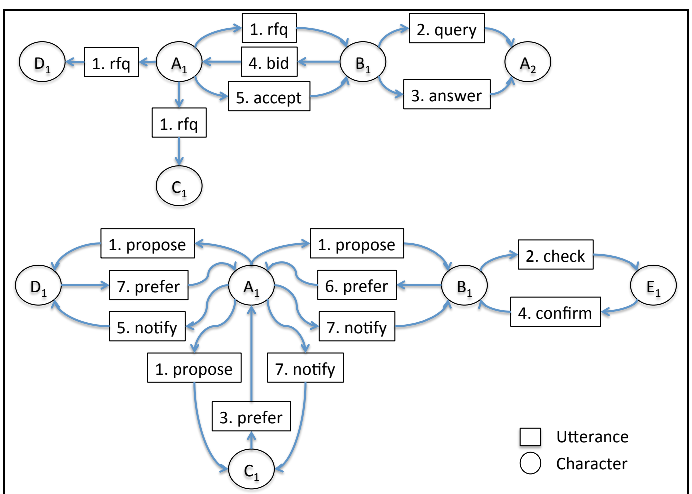

Figure 9: Example Dooley Graphs for purchase negotiation (top) and arranging a meeting (bottom)

A Dooley graph (e.g., Figure 9) is a bipartite graph. One class of nodes (circles in Figure 9) represents *characters*, which are participants at distinguished stages of the discourse, based on the notions of resolution and completion. Thus participant A may appear as nodes A_1 and A_2. The other class of nodes (squares in Figure 9) represents utterances, which are characterized by type of speech act. Utterances that resolve or complete one another tend to form tightly-connected components of the graph, while those that take off in new directions spawn new components. A trajectory through the Dooley graph represents a realization of a conversation. Retrieving a tweet from (say) A to B adds evidence to A-characters and B-characters; recognizing the tweet as a specific speech act adds evidence to utterance nodes requiring that speech act.

We have not yet implemented either the PSE or the EPM on Dooley Graphs. In using a Dooley Graph for surveillance of social media, one would seek to identify well-formed conversations and classify them (e.g., meeting organization, viral propagation of opinion, purchase activities). For this purpose, the EPM should prefer nodes based on their potential for sharpening the distribution over alternative trajectories.

V. Next Steps

Four main avenues for extension of D^{2}REEM provide a range of challenging and important research opportunities: multiple model types, model management, query generation, and model linking.

Multiple Model Types: Our most complete example of D^{2}REEM is the PROPS system, which treats the geospatial domain. The NSM is the next most mature, demonstrating the effectiveness and computational efficiency of modulating the state of a non-geospatial knowledge model by external evidence. In addition to refining these applications, we seek to extend the complete D^{2}REEM cycle to other model types. As we configure the PSE and EPM to different model types, we gain valuable insights into how the underlying mechanisms of the metaheuristic can be generalized.

Model Management: As noted in Section II.A, an important difference between D^2REEM and graph DBs is how knowledge is expressed. Graph DBs construct a small graphical query using the same graph syntax that governs the data, and seek to match the entire query graph against the data graph. D^2REEM uses a large GKM that captures a range of hypotheses, and then explores this model in the light of the data to identify high-priority record-level queries. The use of a complex knowledge model is a strength, in that it externalizes analysts' internal mental models, facilitating collaborative review and enhancement. But it is also a weakness, since constructing such models is itself a labor-intensive process.

For many long-standing problems, model construction is integral to the analytic effort [21], and D^2REEM offers an additional incentive to construct such models. But it will be even more useful if model construction can be partially automated. For example, in the case of the NSM, techniques exist to merge specific narratives in a domain of interest into a NSM [23, 24]. Such technology could exploit past analytic products (which often include a narrative of the event under investigation) to enhance a NSM of the domain. Another example is the Disciple technology [25], which has been used successfully to learn inferential relations of the sort one might encounter in a belief network.

A strength of the PSE approach to model exploration is the locality of ghost movement and pheromone-based interaction. This locality means that GKMs can be extended incrementally, and encourages the notion of a persistent library of dynamically updated models as a central resource in analysis. Development of mechanisms for managing such a library will considerably advance the analytic enterprise.

Query Generation: One task of the EPM is formulating queries that can provide additonal information on GKM nodes that it identifies as highly relevant. In our current implementations, these queries are manually constructed along with the GKM. Given the description of a node in a model and schemata for external data sources, one would like to generate queries automatically, a task that will rely heavily on research in ontological reasoning and semantic web technologies.

Model Linking: The same analytic tasking can be viewed through the lens of multiple model types, and we would like to facilitate the flow of information between these model types by defining mappings between nodes in different model types. Like the previous topic, this one depends on advances in ontological reasoning, as well as model theory and other formal tools [26], and will require attention to aligning multiple levels of meaning [26, 27], not all of which may be represented in each model.

VI. CONCLUSION

Matching knowledge-rich patterns against high-volume, high-velocity data is combinatorially prohibitive. The D^2REEM metaheuristic is a new approach to such retrieval problems that shifts the computational burden from graph matching (a NP-complete problem) to iteratively exploring a knowledge model and issuing focused queries for the data that is most relevant in the light of current knowledge (a process that is linear in the size of the knowledge model). D^2REEM can be applied to any graphical knowledge model in which edges represent causal or or other sequential dependencies and in which adding data to individual nodes can change the probability of a trajectory.

REFERENCES

[1] N. N. Binti Abdullah, "Activity States: a theoretical framework for the analysis of actual human collaboration on the Web," Ph.D., Information, Structure, Systèmes, Université Montpellier II, Montpellier, France, 2006.

[2] H. V. D. Parunak, "Visualizing Agent Conversations: Using Enhanced Dooley Graphs for Agent Design and Analysis," in *Second International Conference on Multi-Agent Systems (ICMAS'96)*, 1996, pp. 275-282.

[3] K. Erol, "Hierarchical Task-Network Planning Systems: Formalization, Analysis, and Implementation," Computer Science, University of Maryland, College Park, MD, 1995.

[4] H. V. D. Parunak, T. Belding, R. Bisson, S. Brueckner, E. Downs, R. Hilscher, and K. Decker, "Stigmergic Modeling of Hierarchical Task Networks," in *the Tenth International Workshop on Multi-Agent-Based Simulation (MABS 2009, at AAMAS 2009)*, Budapest, Hungary, 2009, pp. 98-109.

[5] G. Taylor, B. Bechtel, K. Knudsen, E. Waltz, and J. White, "PSTK: A Toolkit for Modeling Dynamic Power Structures," in *the 16th Behavior Representation in Modeling and Simulation Conference (BRIMS)*, Providence, RI, 2008.

[6] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge, UK: Cambridge University Press, 2010.

[7] H. V. D. Parunak, S. Brueckner, L. Downs, and L. Sappelsa, "Swarming Estimation of Realistic Mental Models," in *Thirteenth Workshop on Multi-Agent Based Simulation (MABS 2012, at AAMAS 2012)*, Valencia, Spain, 2012, pp. 43-55.

[8] P. T. Wood, "Query Languages for Graph Databases," *SIGMOD Record,* vol. 41, pp. 50-60, March 2012.

[9] Neo Technology, *The Neo4j Manual*, 1.8.1 ed.: Neo4j, 2012.

[10] A. Berglund, S. Boag, D. Chamberlin, M. F. Fernández, M. Kay, J. Robie, and J. Siméon. (2011, 22 August). *XML Path Language (XPath) 2.0 (Second Edition)*. Available: http://www.w3.org/TR/xpath20/

[11] S. Harris and A. Seaborne, "SPARQL 1.1 Query Language, W3C Working Draft," W3C2011.

[12] G. Klyne and J. J. Carroll, "Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation," W3C2004.

[13] P. Barcelo, L. Libkin, and J. Reutter, "Querying Graph Patterns," presented at the PODS'11, Athens, Greece, 2011.

[14] M. R. Henzinger, T. A. Henzinger, and P. W. Kopke, "Computing Simulations on Finite and Infinite Graphs," presented at the the □□36th Annual IEEE Symposium on Foun□dations of Computer Science (FOCS 95), Milwaukee, WI, 1995.

[15] M. Kearns, Y. Mansour, and A. Ng, "A Sparse Sampling Algorithm for Near-Optimal Planning in Large Markov Decision Processes," presented at the the Sixteenth International Joint Conference on Artificial Intelligence, 1999.

[16]L. Kocsis and C. Szepesvári, "Bandit based Monte-Carlo Planning," presented at the the EMCL 2006, Berlin, Germany, 2006.

[17]H. V. D. Parunak, S. A. Brueckner, R. Matthews, and J. Sauter, "Swarming methods for geospatial reasoning," *International Journal of Geographical Information Science,* vol. 20, pp. 945-964, 2006.

[18]H. V. D. Parunak, "Real-Time Agent Characterization and Prediction," presented at the International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'07), Industrial Track, Honolulu, Hawaii, 2007.

[19]B. Horling, V. Lesser, R. Vincent, T. Wagner, A. Raja, S. Zhang, K. Decker, and A. Garvey. (2004, 1 August). *The Taems White Paper* [Web site]. Available: http://dis.cs.umass.edu/research/taems/white/

[20]S. Brueckner, T. Belding, R. Bisson, E. Downs, and H. V. D. Parunak, "Swarming Polyagents Executing Hierarchical Task Networks," in *Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2009)*, San Francisco, CA, 2009, pp. 51-60.

[21]L. Sappelsa, H. V. D. Parunak, and S. Brueckner, "The Generic Narrative Space Model as an Intelligence Analysis Tool," *American Intelligence Journal,* vol. 31, p. (in press), 2013.

[22]P. R. Cohen and C. R. Perrault, "Elements of a Plan-Based Theory of Speech Acts," *Cognitive Science,* vol. 3, pp. 177-212, 1979.

[23]B. Li, S. Lee-Urban, D. S. Appling, and M. O. Riedl, "Crowdsourcing Narrative Intelligence," *Advances in Cognitive Systems,* vol. 1, pp. 1-18, 2012.

[24]M. O. Riedl and R. M. Young, "From Linear Story Generation to Branching Story Graphs," *IEEE Computer Graphics and Applications,* vol. 26, 2006.

[25]Gheorghe Tecuci, M. Boicu, C. Boicu, D. Marcu, B. Stanescu, and M. Barbulescu, "The Disciple-RKF Learning and Reasoning Agent," *Computational Intelligence,* vol. 21, pp. 462-479, November 2005.

[26]A. Tolk, S. Y. Diallo, and J. J. Padilla, "Semiotics, entropy, and interoperability of simulation systems: mathematical foundations of M&S standardization," presented at the Proceedings of the Winter Simulation Conference, Berlin, Germany, 2012.

[27]R. Ackoff, "From Data to Wisdom," *Journal of Applied Systems Analysis,* vol. 16, pp. 3-9, 1989.