Semantic Technology for Intelligence, Defense and Security

STIDS 2011

Building the Semantic Cloud

George Mason University

Center of Excellence in Command, Control, Communications, Computing and Intelligence

Northern Virginia AFCEA Chapter

Conference held at:

The Mason Inn
George Mason University
Fairfax, Virginia Campus
16 - 17 November 2011

Online Proceedings

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

STIDS 2011

Preface

The 6th International Conference on Semantic Technologies for Intelligence, Defense, and Security (STIDS 2011) 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. This year we have the following topics:

- 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 2011.

Paulo Costa and Kathryn Laskey
STIDS 2011 Chairs

CONFERENCE CO-CHAIRS

Name	Location
Paulo Cesar G. Costa	George Mason University
Kathryn Blackmond Laskey	George Mason University

SCIENTIFIC COMMITTEE

Name	Affiliation
Bill Andersen	Ontology Works
Rommel Carvalho	George Mason University
Werner Ceusters	University at Buffalo
Paulo Costa	George Mason University
Tim Darr	Knowledge Based Systems, Inc.
Katherine Goodier	NCI, Inc.
Kathleen Hornsby	University of Iowa
Terry Janssen	SAIC
Kathryn Laskey	George Mason University
Nancy Lawler	US Department of Defense
Dan Maxwell	KaDSci, Inc.
Fabian Neuhaus	NCOR
Leo Obrst	MITRE Corporation
Mary Parmelee	MITRE Corporation
Marvin Simpson	Optech, Inc.
Barry Smith	NCOR, University at Buffalo
Gheorghe Tecuci	George Mason University
Sandra Thompson	US Department of Energy
Andreas Tolk	Old Dominion University
Brian Ulicny	Vistology, Inc.
Duminda Wijesekera	George Mason University

STIDS 2011 Platinum Sponsor

Data Tactics Corporation (DTC) has been developing and implementing mission-focused capabilities to the Intelligence Community and DOD for years; providing superior service and leading innovation. Whether it is data mining, data correlation, data retrieval, information security or cloud computing, Data Tactics understands the challenges that face our client-base and our peers across the industry. With our vast knowledge, professional expertise and dedication, Data Tactics is prepared and committed to designing, implementing and sustaining customized solutions to meet the customers' mission requirements.

Data Tactics Corporation is a small business solely focused on mission –relevant solutions that bring industry recognized experts in the field of Specialized Cloud Enterprise Architecture, Cyber Security, Geospatial Engineering, System / Software Development, Data / System Integration, and Operations and Maintenance (O&M) / Sustainment support. We measure that support at our end-user. The staff is qualified to identify report, resolve and support a myriad of complex data, storage, security and system problems. Our success has been proven time and again at traditional sites but also in tactical forward deployed environments.

Our Mission

- To Design, Develop, Deliver and Manage State-of-the-Art Technological Capabilities for Our Client's Enterprise that Supports Our Client's Mission Objectives
- To See Our Performance across the Service Lifecycle through Our Client's Lens.
- Our Work Contributes to Our Client's Success because we Design, Deliver and Sustain those Services to Work in the Client's Environment, by Client's Personnel to Achieve Client Success

Vision Statement

- To Establish an Enduring Relationship of Trust with Our Client Based Solely on Our Performance
- To Deliver a Product or Service that becomes Second-Nature to Our Client's Personnel and a Seamless Part of Our Client's Business Operations
- To Remain a Creative, Disruptive and Leading Research, Development and Rapid Deployment Institution Where Our Shared Intellect, Hard Work and Vanguard for Our Client's Trust make a Positive Difference in the Lives of Our Employees, the Success of Our Clients and the Security of Our Country

Technical Papers

Integration of Intelligence Data through Semantic Enhancement

Salmen David
dsalmen@data-tactics.com
Data Tactics Corp.

Malyuta Tatiana
tmalyuta@data-tactics.com
Data Tactics Corp.,
City University of New York

Hansen Alan
alan.hansen1@us.army.mil
Intelligence and Information Warfare Directorate

Cronen Shaun
shaun.cronen@us.army.mil
Intelligence and Information Warfare Directorate

Smith Barry
phismith@buffalo.edu
National Center for Ontological Research, University at Buffalo

***Abstract*—We describe a strategy for integration of data that is based on the idea of semantic enhancement. The strategy promises a number of benefits: it can be applied incrementally; it creates minimal barriers to the incorporation of new data into the semantically enhanced system; it preserves the existing data (including any existing data-semantics) in their original form (thus all provenance information is retained, and no heavy pre-processing is required); and it embraces the full spectrum of data sources, types, models, and modalities (including text, images, audio, and signals). The result of applying this strategy to a given body of data is an evolving Dataspace that allows the application of a variety of integration and analytic processes to diverse data contents. We conceive semantic enhancement (SE) as a light-weight and flexible process that leverages the richness of the structured contents of the Dataspace without adding storage and processing burdens to what, in the intelligence domain, will be an already storage- and processing-heavy starting point. SE works not by changing the data to which it is applied, but rather by adding an extra semantic layer to this data. We sketch how the semantic enhancement approach can be applied consistently and in cumulative fashion to new data and data-models that enter the Dataspace.**

Keywords: integration, intelligence data, ontology, semantic technology.

I. INTRODUCTION

The success of the war fighter and homeland defender in the Net-Centric Warfare environment is largely defined by the ability to quickly acquire and efficiently and accurately process intelligence information from numerous heterogeneous sources of different structure and modality. Traditional data integration approaches fail in the face of the scale, diversity, and heterogeneity of intelligence data sources and data-models because they fail to address one or more of the following requirements:

- Integration must proceed without heavy pre-processing
- Integration must proceed regardless of the data-models used (or not used) in the data sources to be integrated,
- Integration must proceed regardless of the data modality, and without loss or distortion of data, of its associated data semantics, and of data-provenance information,
- Integration must involve the ability to incorporate multiple points of view on the data to be integrated, including different views of the data, for example on the part of different analysts using different analytical tools.

As a first step towards meeting these requirements we introduced in 2009 the Data Representation and Integration Framework (DRIF) [1, 2], which presents minimal barriers to the incorporation of new data into a data resource, thus requiring no heavy pre-processing and no data or data-model conditioning. DRIF embraces the full spectrum of data sources, types, models, and modalities, including text, images, audio, and signals, while supporting a variety of integration and analytic processes and tools. Details are presented below.

The Dataspace store of intelligence data which is the subject of this communication is the result of applying the DRIF to the task of integrating very large heterogeneous primary data artifacts. As the Dataspace has evolved through time, so it has incorporated progressively ever larger quantities of data, and also more specific local implementations and data structures used by data analysts, some of which bring their own data semantics. For the purposes that the Dataspace is intended to serve, it is vital that no restrictions are imposed either on the types of source-artifacts and the associated models and media within the Dataspace, or on the processes by which the Dataspace is populated (whether by loading structured data from a database, by extraction from a text document through some Natural Language Processing application, by automatic analysis of signals, or through inference by a human analyst).

The design of the Dataspace is such that it can incorporate hundreds of millions of unstructured documents and similarly large quantities of images, signals data, and other structured and unstructured primary data artifacts. Each of these artifacts, when it enters the Dataspace, is represented through a set of metadata, including labels specifying image type, MIME type, and so forth, as well as provenance information. Further processing may, for example, associate pixels in an image with the name of a person, or a range of characters in an unstructured text document with the name of a location, or extract a cell from a database table. The DRIF provides a common framework in which the results of all of these processes are represented in a unified way, details of which are provided below. As a result, primary data can be utilized immediately upon entering the Dataspace for a variety of different kinds of search and more sophisticated processing based thereon. DRIF is not, however, a magic bullet; many issues of data integration at the syntactic level will remain, arising for example as a result of data formats which do not match, where we will need to normalize the format into an augmented model that will serve as the target of annotations. This will involve considerable effort to ensure that the needed actions are performed promptly and consistently whenever new data comes in. Here, however, we focus exclusively on those issues which arise at the stage of what we can loosely call the 'representational' aspects of data integration.

Some primary artifacts within the Dataspace already incorporate useable semantic content – for instance a structured database which incorporates meaningful column headers, or a message with a structured payload incorporating meaningful tags. But such content is *ad hoc*. It is tied to specific local implementations and typically falls short of what is needed to secure semantic interoperability of the implementations involved because of the absence of a common formally coherent approach to semantics and of a common governance process.

Moreover, full semantic integration is in any case prevented by the needs of openness of the Dataspace to ever new sorts of primary data and analytically derived data. It is to compensate for this problem that we have developed our strategy for semantic enhancement. We start out from the assumption that semantic data enrichment can be achieved only incrementally, through the step-by-step creation of ontology modules that are designed in coordinated fashion to work well both with each other and with specific bodies of Dataspace content. The vision is a lightweight, flexible approach comprising an *extra ontology layer* that leverages the contents of the Dataspace without adding storage and processing weight to what is an already storage- and processing-heavy resource. We discuss the details of semantic enhancement in section IV. First, however, we introduce the DRIF and the Dataspace to which the SE strategy will be applied.

II. Data Representation and Integration Framework

Our starting point is a body of U.S. Department of Defense (DoD) intelligence data within what we are here calling the Dataspace. The implementation in the specific context upon which we focus here is engineered around cloud computing paradigms and is primarily based upon open-source cloud software stack components. This cloud computing foundation leverages advantages of linear scaling and parallel distributed computation when faced with the reality of ever increasing data volumes and integration processing. All the work described is either deployed or in the final stages of testing prior to deployment.

The Dataspace is built using the Data Representation and Integration Framework (DRIF), which has been designed to represent large quantities of data in a form that is useful to the end user both for direct inspection and for the application of various kinds of analytics. Representations of source data artifacts and their contents within the Dataspace are of two forms, which we call *primary* and *derived*, respectively. The Dataspace is divided into corresponding segments (see Figure 1) in a way that supports a comprehensive approach to integration that allows accommodation of the multiple views of the primary and derived data and of the associated data-semantics and metadata which arise for example as a result of the workings of multiple different sorts of analytical tools.

A. Approach to Integration

Our approach to integrating intelligence data starts with source artifacts consisting of primary data across a variety of representation modalities. This primary data is weakly integrated in the sense that indexes are provided to support simple (string-based) data search across all primary artifacts.

Some primary data comes with its own native structure, and further structure will typically be added thorough analytical processing. The second integration step addresses the need for the unified storage of this structured data to support more complex structured search across both primary and derived artifacts.

Importantly, we here embrace the diversity of domain-specific data-models employed throughout the Intelligence Community while at the same time reaping benefits from an approach that is data-model agnostic. This is because the unified representation provided by the DRIF allows analytic processing of data in highly diverse primary artifacts associated with different native data-models to be used as targets of cross-artifact analytics. For example, and most simply, it is possible to perform unrestricted string search across structured artifacts of highly different sorts. Examples of more sophisticated analytics include computer-aided data-model harmonization, for example by allowing significant overlap of sets of values of attributes from different databases to be flagged by the analytic process as a potential indication that the attributes have the same meaning, thereby making it possible for the relevant portions of the two databases to be enriched through fusion.

B. Dataspace Organization

The organization of the Dataspace is schematically illustrated in Fig. 1.

Segment 0 is a store of primary artifacts, including documents, images, signals, and analysts' work products vetted for re-use as input for further processing. The physical implementation of Segment 0 may be such that all data is stored internally; or it may be distributed, so that source artifact data may for example be either contained in the cloud store or stored externally to the Dataspace and referenced in the cloud store. Primary data vary widely by nature; they may have different structures (for example of a relational database), or they may be unstructured (for example, free text, audio or video files), and they may be of different modalities (for example they may be cells of a relational database, audio sequences, assertions of an analyst).

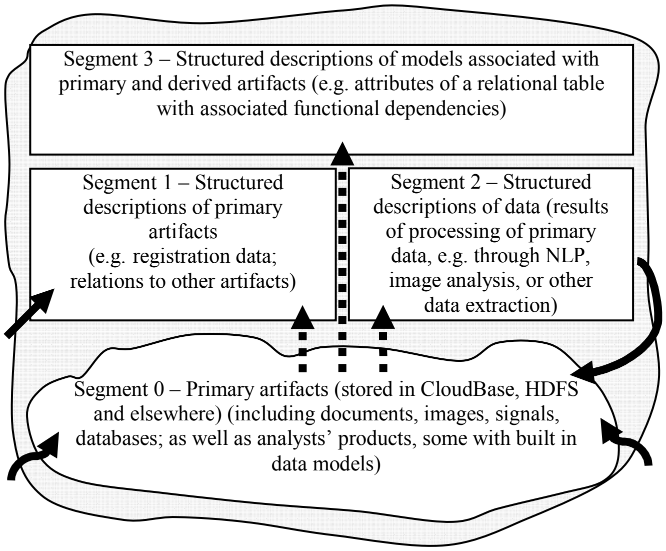

Segment 3 – Structured descriptions of models associated with primary and derived artifacts (e.g. attributes of a relational table with associated functional dependencies)
Segment 1 – Structured descriptions of primary artifacts (e.g. registration data; relations to other artifacts)
Segment 2 – Structured descriptions of data (results of processing of primary data, e.g. through NLP, image analysis, or other data extraction)
Segment 0 – Primary artifacts (stored in CloudBase, HDFS and elsewhere) (including documents, images, signals, databases; as well as analysts' products, some with built in data models)

Figure 1. Organization of the Dataspace. Solid line: registration processes; curved solid lines: processes that ingest artifacts into the Dataspace, including feeding back into the Dataspace analysts' products – results of the Dataspace processing ; dashed lines: derivation processes.

Segment 1 includes primary artifact registration data as well as specifications of relations between artifacts (for example, nesting of an image within a document, or attachment of one document to another). Segment 1 will include also data pertaining to the way each derived artifact of Segments 2 and 3 is derived from primary artifact(s) in Segment 0.

Segment 2 stores the structured data that is either already present in primary artifacts or derived therefrom through analytic processing resting on data-models represented in Segment 3.

Segment 3 stores the descriptions of the data-models used in Segment 2. These data-models may include database schemas, message formats, or XML schemas. The data-models themselves are primary artifacts and are thus stored in Segment 0 and registered in Segment 1.

The Dataspace is evolving continuously not only because of new primary data ingested from the outside, but also because new artifacts are being created, for example, through analysts' reports based on processing of existing data. These artifacts themselves have a status of new primary artifacts.

C. Segments as Abstractions Over the Artifacts

Each of Segments 1-3 is an abstraction over the corpus of primary data artifacts (Segment 0) and supports analytics of a particular type:

- Segment 1 is a high-level view of the entire artifact corpus including the relations between the artifacts, but with no reference to their internal contents.
- Segment 2 is a collection of detailed views of the internal contents of the artifacts at the level of individual data items.
- Segment 3 describes the data-models which support the two sets of views just mentioned as well as synoptic views (ultimately including SE-based views) of the type which can foster harmonization.

D. Where Models and Primary Data Come Together

We believe that the principal contribution of the Dataspace endeavor is to resolve certain problems of storage and thus of representation, enrichment, and evolution of large bodies of data. The goal is to provide room for both primary data and the multiple results of processing these data by different analysts or analytic methods. To achieve this we introduced in [3] a strategy for description of data that is designed to enable true data integration across a constantly evolving and highly heterogeneous resource comprehending extremely large volumes of data. As already recognized at the very beginning of contemporary high-level research in biomedical ontology [4], this end can be achieved only if data are exposed in a way that is independent of their original intended use. This must involve some means to represent original data-models at a level of abstraction that is higher than that of primary data. We accordingly propose an abstract data-model based on five core elements: sign, concept, term, predicate, and statement, which we believe is sufficient to represent any data-model in these terms.

Sign: A *sign* g_i is a string that is the abstracted proxy within the dataspace for one or more chunks of data used in some primary artifact with the intention of referring to some individual entity (e.g. person, location, organization, object, event). Examples include: a sign of the type proper name that is associated with an expression (for example 'he' or 'Dr. Watkins' occurring in a document; a label annotating an area in a pixel array as forming an image of some building; a label annotating a fragment of an audio stream or other signal as recording some explosion event. Each sign is associated with one or more physical extents within those primary artifacts with which it is associated, which we call *mentions* (the latter are what are elsewhere called *tokens*). The collection $G = \{g_i\}$ comprehends all signs extracted from primary data artifacts and changes with the incorporation of new artifacts.

Concept: A *concept* c_i is (for the purposes of this exposition) a string that is used in the Dataspace to represent some general category or grouping. The purpose of concept strings is to represent and allow reuse of classifications native to primary artifacts. Concepts are taken from data-models registered in Segment 1. Examples of concepts are: the classes of an ontology such as UCore SL, the tag set in an XML Schema Document (XSD), and the attribute or table names in a relational database. The collection $C = \{c_i\}$ comprehends all concepts within the Dataspace and changes as new data-models are incorporated.

Term: A *term,* t_{ij}, is an ordered pair of strings $<g_i,c_j>$, where $g_i \in G$ and $c_j \in C$. Each term results from a process of contextual disambiguation of a *sign*, a process which associates a *sign* with a *concept*, as in <123-45-6789, SSN>. The collection $T =$

$\{t_{ij}\}$ comprehends all terms identified by analytic processing of primary artifacts.

Predicate: A *predicate* (by which we mean here always: *binary relational predicate*) p_i is a string that is used to connect terms in accordance with domain and range constraints. Predicates are used in the formation of *statements* (as described below). Examples of predicates are: hasSSN, hasLocation, hasBirthDate. Predicates are derived from data-models registered in Segment 1, for example from table column headings or from XML tags. The collection $P = \{p_i\}$ comprehends all predicates within the Dataspace and changes as new data-models are added.

Statement: A *statement* s_i is an ordered triple consisting of a subject, a predicate, and an object. The collection $S = \{s_i\}$ of statements is recursively defined. At the lowest level, statements are ordered triples consisting of a term, a predicate, and a second term. In higher-level statements, subjects and objects may be lower-level statements. Examples: <[Bruno, PersonName] hasSSN [123-45-6789, SSN]>

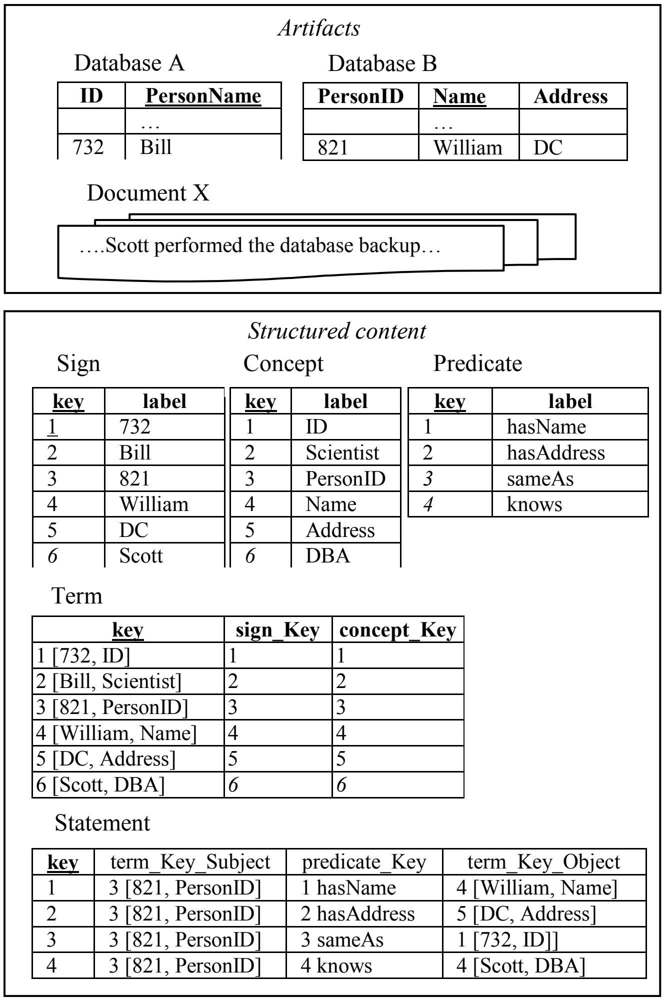
The five primitives of the DRIF (sign, concept, predicate, term, and statement) define a data reference model which, by effectively decoupling data from data-models, can represent any sort of data-model at the level that is useful for integration.

Fig. 2 schematically illustrates the representation of structured data in accordance with the DRIF for three sample primary artifacts, two of them relational databases, the third an unstructured document. The example also shows how data-semantics come to be added to the Dataspace in *ad hoc* fashion – here, because an analyst decides to to introduce a new Concept DBA (meaning: database administrator). Additional Statements establishing relationships between Terms using Predicates SameAs and Knows are also included in the Figure.

The reader familiar with the Resource Description Framework (RDF/RDFS) may wonder what is different here. RDF employs a similar level of abstraction, but it is a language, while what we are offering here is a specific, albeit still highly abstract, data-model. This data-model could of course be specified very easily using the RDF language; but it could be specified also using relational database or some other storage technology. Our choice of data-model was motivated further by the fact that our implementation and security requirements dictated the use of a specific type of cloud storage solution [5, 6] that is both highly scalable and offers highly granular security access controls.

III. Semantic Enhancement

The DRIF focuses on the representational aspects of the Dataspace and on the basic types of data integration that such representation provides. In what follows we describe the current phase of evolution of DRIF, the phase of Semantic Enhancement (SE). SE, as we conceive it, is a light-weight and flexible solution that leverages the richness of the native source data and of any local semantics associated with these data without adding storage and processing weight. The SE strategy is compliant with and complements the DRIF.



Artifacts

Database A

ID	PersonName
	…
732	Bill

Database B

PersonID	Name	Address
	…	
821	William	DC

Document X

Structured content

Sign

key	label
1	732
2	Bill
3	821
4	William
5	DC
6	Scott

Concept

key	label
1	ID
2	Scientist
3	PersonID
4	Name
5	Address
6	DBA

Predicate

key	label
1	hasName
2	hasAddress
3	sameAs
4	knows

Term

key	sign_Key	concept_Key
1 [732, ID]	1	1
2 [Bill, Scientist]	2	2
3 [821, PersonID]	3	3
4 [William, Name]	4	4
5 [DC, Address]	5	5
6 [Scott, DBA]	*6*	*6*

Statement

key	term_Key_Subject	predicate_Key	term_Key_Object
1	3 [821, PersonID]	1 hasName	4 [William, Name]
2	3 [821, PersonID]	2 hasAddress	5 [DC, Address]
3	3 [821, PersonID]	3 sameAs	1 [732, ID]]
4	3 [821, PersonID]	4 knows	4 [Scott, DBA]

Figure 2. Simplified example of structured content derived from 3 primary artifacts.

A. Goals of Semantic Enhancement

SE is a strategy that is currently being implemented to improve our handling of the enormous heterogeneity of Dataspace content. It is centered on building a flexible and extensible framework of hierarchically organized, controlled structured vocabularies – called 'ontologies' – covering different areas of relevance to intelligence analysis. The framework will be constructed in part by reusing already existing resources, in part through collaboration with other defense and military organizations in the creation of new ontology modules. The ontologies will be used in an incremental process of annotation (or 'tagging') of those concepts and predicates already identified in data-models within the Dataspace along the lines described in our discussion of Segment 3 above. The latter amount to what we referred to above as '*ad hoc* semantics'. Because the salient data-models derive from so many heterogeneous sources, they use a multiplicity of partially overlapping and partially conflicting vocabularies, which it is the task of SE to reconcile by associating co-referring concepts and predicates (strings) employed within distinct data-models in the Dataspace to single nodes within the external SE ontologies.

To function in the needed way, annotations must be cumulative, in the sense that our strategy will ensure that tags created by different annotators will be consistent with each other. The value of annotations must also be preserved when the SE ontologies change, for example through refinements created to reflect advances in knowledge, and to this end the ontologies must be subject to strict versioning policies.

Finally, the SE framework must be implemented in such a way that it can serve not merely as a tool of harmonization of the data-models internal to the Dataspace but also in a way that allows integration with other, external data resources wherever common ontologies are used for annotation.

To address these constraints is by no means a simple matter. When data value codifications do not match – for example when we have 1,2,3 in one data source, R, G, B in another data source, and RED, GREEN, BLUE in our Color ontology, then annotation for each source to hierarchy values can be very labor intensive and require significant SME effort.

B. Sample Benefits of Semantic Enhancement

We can see the sorts of benefits that SE will provide already at the level of search, where problems arise because of the multiple different ways of describing data within the Dataspace. Problems that need to be confronted include:

1. The need to find data items identified by means of terms which are *narrower* or *broader* in meaning than the terms analysts will standardly use when searching;
2. The need to find data items in documents that are formulated using a language or technical jargon with which analysts are unfamiliar.

To provide some very simple examples: we know that a given package 'has been shipped with a red label', but the documents that we have pertaining to this package use only the word 'vermillion'; or we need to find references to a package identified as 'containing furniture', but the documents we have refer only to 'chairs'; or we need to find a given package suspected of containing crack cocaine, but the audio recordings we have at our disposal relating to this package refer only to 'bobo' or 'botray' or 'boubou'. If we are restricted to string search, our queries would not return the needed results. Hence, we need a framework which expands string search by capturing type and subtype information, and also incorporates synonym information. These needs are targeted along two dimensions; first, through the fact that all SE ontologies will be organized around a central backbone subtype (or *is_a*) hierarchy; and second through the progressive incorporation in all nodes of the SE ontologies of links to relevant synonyms derived through the annotations which will link ontology nodes to the rich collection of corresponding concepts and predicates in other areas of the Dataspace.

C. The Strategy for Semantic Enhancement

Our strategy is designed to achieve its goals not by changing the Dataspace, but rather by adding an extra *semantic layer* thereto. The strategy is thus similar to that underlying the Universal Core (UCore), which arose out of the National Information Sharing Strategy supported by multiple U.S. Federal Government Departments, by the intelligence community, and by a number of other national and international organizations [7, 8]. Here, a small controlled vocabulary was provided for multi-community use to associate simple summary tags to message payloads for purposes of data search and integration.

Reflecting the extreme diversity of intelligence data, multiple subject-matter expert communities will be contributing to the SE. For the strategy to work and provide useful and efficient integration, these multiple distributed teams must use the SE approach in a consistent fashion. Previous efforts to create a broad-based, multi-community ontological approach to data integration in defense and intelligence domains have failed because the incompatible, and often over-simplistic, views of reality incorporated into legacy databases and data-models led to incompatible development of ontologies in ways that precluded interoperability. Many advocates of semantic approaches to data integration have still failed to appreciate the tremendous challenges, both technical and human, created by the entrenched predisposition on the part of ontology developers to create ontologies each on the basis of their own potentially idiosyncratic data representations.

The solution which we advocate is modeled on the successful semantic annotation approach pioneered in the field of bioinformatics by the Gene Ontology [9]. This approach is now being pursued systematically within the framework of the OBO Foundry [10, 11], which starts out from the idea that the most effective way to ensure mutual consistency of ontologies created by multiple independent groups over time and to ensure that these ontologies are maintained in such a way as to keep pace with advances in knowledge is to organize ontologies as a collection of modules with discrete (non-overlapping) subject-matters maintained by subject-matter experts, according to a strategy outlined in [12]. To ensure consistency, these ontologies should be created as extensions of more generic higher level ontologies, subject to common rules for example concerning the treatment of definitions, and they should be based on a small common upper-level ontology (ULO), whose domain and content neutral. For example, it will include relations such as *is-a* (for subtype), *member-of*, *part-of*, and so on. As initial ULO we choose the Basic Formal Ontology (BFO) [13], which has been implemented in more than 100 similar projects, and which serves as the basis of the already mentioned UCore Semantic Layer [8].

The ULO will be associated with a small number of Mid-Level Ontologies (MLOs) defined by downward population from the ULO. The MLOs will serve in turn as bridge to a number of Low-Level Ontologies (LLO), which will specify narrow content domains. Each MLO represents cross-domain entities, such as Person or Information, and will be constructed in tandem with the LLOs which it subsumes in order to ensure the mutual consistency and interoperability of the subsumed LLOs. The MLOs and LLOs must in turn be associated with the resources of a relation ontology, providing for the representation of content-specific relations such as Owns, WorksFor, Audits, and so on.

Initial due diligence efforts in our strategy of semantic enhancement requires us to identify an initial collection of authoritative codifications at Mid- and Lower Levels – along roughly the lines depicted in Table 1 – and to begin the process of formalizing them within the BFO common upper-level ontological framework. In some areas ontologies will need to be created *de novo*, since no adequate authoritative codifications will exist.

Examples of MLO cross-domains

- Geospatial
- Biometrics
- Person
- Provenance and Trust
- Organization
- Signals and Sensors
- Equipment
- Facility

Examples of LLO domains

Subsumed by Geospatial

- Geospatial Feature
- Country

Subsumed by Biometrics

- Fingerprint
- Iris

Subsumed by Person

- Employment Data
- Criminal Data
- Medical Data
- Ethnicity and Tribe
- Skill

Subsumed by Provenance and Trust

- Data Quality
- Access Permissions
- Data Source
- Evidence

Table 1: Sample Ontologies within the SE Structure

D. *Implementation of the SE Strategy*

We can now outline the steps which are involved in realizing this strategy in the specific context of the Dataspace, where we already have data structured using the DRIF.

First Step: Review the contents of the Dataspace, specifically that concepts and predicates in Segment 1, and identify a subset of topic areas where data integration is a priority for analytics.

Second Step: Formulate a list of MLOs that would be needed to annotate the data in corresponding areas. As far as possible identify existing ontologies which may potentially be reused for this purpose, and build initial versions of new ontologies where needed.

Third Step: Identify a specific subset of the content of the source data-models, and identify LLOs that will capture this subset in a semantically coherent fashion, ensuring that each LLO is subsumed by some MLO. Subject matter experts should be recruited to take charge of creation and maintenance of the LLOs and MLOs and of their use in annotations. In this way we can create a cadre of SMEs with expertise in annotation and in supporting semantic enhancement.

In realizing the above we need to maximize as far as possible the *reuse* of ontologies which are already being used by relevant communities. This is because the strategy will be successful only to the degree that a critical mass of potential users are able to be convinced of its utility and thus incentivized to engage in advancing it further for example by extending it new types of data and by disseminating the resource to new groups of analysts. Reusing already existing ontologies will not merely provide a core of familiar terms which analysts can use for search purposes, it will also increase the degree to which we can integrate into the Dataspace data that has already been annotated in consistent fashion by external bodies.

Fourth Step: When once a stable, initial set of ontologies has been created, we use these ontologies to annotate the data-models in corresponding portions of the Dataspace. As should by now be clear, the entire strategy is an incremental one, based on a principle of low hanging fruit: the idea is not to import the above ontologiesas a whole; rather we examine the existing Dataspace resources and identify expressions therein for which counterparts in the ontologies already exist or can easily be added. In constructing the ontologies these expressions will be provided with a common logical architecture and a common set of relations defined through the ULO top level and in terms of which logical definitions for terms in the ontologies can then be formulated. The result can be used as a basis for the application of general-purpose tools, including standard OWL reasoners FaCT++, RACER, or Pellet, which can be used to check ontologies in the SE resource for mutual consistency.

Stage 4 of the SE process consists in associating each set of equivalent data source concepts with a single common MLO or LLO expression (which will be added at the appropriate level within the SE ontology structure where not already present). Further types of integration are thereafter brought about automatically. Whenever any Dataspace resource becomes linked to one of our chosen ontologies in a way that can be used to generate corresponding annotations, it thereby becomes linked to all the other Dataspace (and external) resources that have already been annotated with the same SE ontologies. This creates a snowball effect, whereby each new annotation increases the value of existing annotations [9], and provides further incentives for the use of the SE ontologies by new groups of users.

E. Organization of the SE Ontologies

Fig. 3 illustrates the organization of the SE ontology space. Each LLO represents the reality of a particular narrowly defined domain, for example in an area such as Education and Skills.

An MLO is a container of LLOs. Since we will be developing LLOs in step-by-step fashion to address what are at any given time the most urgent needs of Dataspace users, there will be data which cannot as yet be annotated with the full granularity of detail which the annotator requires. The strategy is to use such cases to advance the further development of the ontology resource base, again following the model tested in the bioinformatics domain [9]. For example, an analyst may want to use the SE resources to extract and disambiguate data from a particular document. For different reasons the analyst may not be able to use the most detailed semantics and will use a more general one. LLO taxonomies will also be used by analytics to produce results of different level of detail: from a fine-grained view of narrow areas within the Dataspace to coarse grained pictures of larger domains.

Because original data and data-semantics are in every case preserved without loss or distortion in the Dataspace as it exists prior to Semantic Enhancement, there is no need to represent all details of original storage data structures in the SE stage. This means that complex ontologies are not needed – a common and shared vocabulary is sufficient for virtual semantic integration and search/analytics, while underlying details are maintained by the authors of specific primary artifacts. Similarly, the collection of SE ontologies does not need to cover all of the *ad hoc* local semantics within the Dataspace – content that is unlikely to be used in search or is not important for integration can be excluded from the Enhancement step, since it will still be available in the source data-models and can be accessed when drilling down to the appropriate level.

The SE approach is highly flexible. It represents a "pay-as-you-go" approach in the sense that investments can be made only in specific areas according to identified need. It is also tunable in the sense that, if a given body of annotations for a particular subset of a source data-model is too general for data analyst purposes, then the respective LLOs can be further developed as needed.

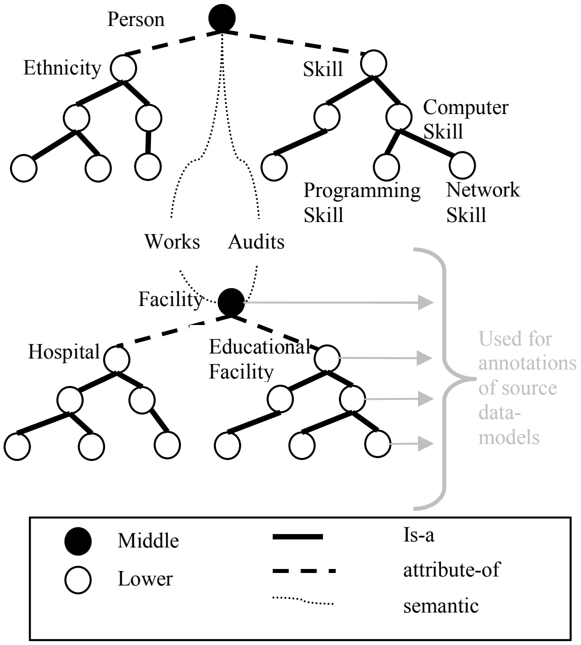


Figure 3. Simplified Example of an SE Ontology Structure.

IV. Conclusion

Together, the DRIF and SE provide what we believe is a workable data-integration solution. The DRIF is a highly flexible framework, with few constraints and including an RDF-style decomposed representation of structured data which allows the collection of data resources without loss or distortion in a way that achieves syntactic integration and preserves the local semantics of primary sources and of analytics software. SE provides semantic integration in a light-weight yet incrementally extendible fashion, and in a way that can foster global integration without adding storage and processing weight to already storage- and processing-heavy Dataspace.

The SE approach provides a strategy to allow the Dataspace to be understood as evolving *cumulatively* as it accommodates new kinds of data. It provides a more *consistent, homogeneous,* and *well-articulated* representation of structured content that originates in multiple internally inconsistent and heterogeneous models. And while it involves considerable initial SME investment in ontology creation and annotation, we believe that it will allow the management and exploitation of the Dataspace to become more *cost-effective* over time.

In addition, the use of the selected MLOs and LLOs brings integration with other government initiatives and brings the Dataspace endeavor closer to the federally mandated net-centric data strategy; it also makes the integrated Dataspace more effectively searchable and provides an expanding body of content to which more powerful analytics can be applied in the future.

Acknowledgments: This work was funded by US Army CERDEC I2WD. The authors thank Mr. Kesny Parent, DCGS-A Branch Chief, for continued support.

REFERENCES

[1] S. Yoakum-Stover, T. Malyuta, N. Antunes, "A Data Integration Framework with Full Spectrum Fusion Capabilities", Presented at the Sensor and Information Fusion Symposium, Las Vegas, NV, Aug 3-7, 2009.

[2] A. Hansen, D. Salmen, T. Malyuta, and N. Antunes. "An Evolving Integrated Dataspace on the Cloud." Presented at the Sensor and Information Fusion Symposium, Las Vegas, NV, July 26-29, 2010.

[3] S. Yoakum-Stover, T. Malyuta, "Unified Integration Architecture for Intelligence Data", Proceedings of DAMA International Europe Conference, London, UK, 2008.

[4] Rosse, C. and Mejino, J. L. V. A Reference Ontology for Bioinformatics: The Foundational Model of Anatomy. Journal of Biomedical Informatics 36, 2003, 478-500.

[5] R6 Cloudbase documentation and source code.

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

[7] http://ncor.us/ucore-sl.

[8] B. Smith, L. Vizenor and J. Schoening, "Universal Core Semantic Layer", Ontology for the Intelligence Community, Proceedings of the Third OIC Conference, George Mason University, Fairfax, VA, October 2009, CEUR Workshop Proceedings, vol. 555.

[9] D. Hill, et al., "Gene Ontology Annotations: What they mean and where they come from", BMC Bioinformatics, 2008; 9(Suppl 5): S2.

[10] B. Smith, et al., "The OBO Foundry: Coordinated Evolution of Ontologies to Support Biomedical Data Integration", Nature Biotechnology, 25 (11), November 2007, 1251-1255.

[11] B. Smith, W. Ceusters "Ontological Realism: A methodology for coordinated evolution of scientific ontologis", Applied Ontology 5 (2010) 139-188, http://x.co/adRJ.

[12] W. Ceusters, B. Smith, J. M. Fielding, "LinkSuite^{TM}: formally robust ontology-based data and information integration," in Database Integration in Life Sciences, Berlin, Springer, 2004. http://ontology.buffalo.edu/bio/LinkSuite.pdf

[13] Basic Formal Ontology. http://www.ifomis.org/bfo/.

SCUBA: An Agent-Based Ontology Creation and Alignment Method for Socio-Cultural Modeling

Donald R. Kretz
Raytheon IIS
Garland, TX
donald_r_kretz@raytheon.com

William D. Phillips
Raytheon NCS
Largo, FL
bill.phillips@raytheon.com

Bruce E. Peoples
Raytheon IIS
State College, PA
bruce_e_peoples@raytheon.com

Abstract -- **An otherwise promising business, political, or military strategy can be crippled by an incomplete understanding of the social-cultural factors that define and influence a region. Such omissions are sometimes due to oversight, but often stem from a fundamental lack of understanding of how to model such difficult and unfamiliar concepts. The information required to generate useful contextual models is typically available but vast, and manual interpretation of detailed text is time-consuming, highly subjective, and requires specialized skills. The SCUBA project achieved a balanced human-computer modeling paradigm to 1) automate the creation of social and cultural ontologies from selected source materials using previously-developed tools, 2) apply a variety of nominal, semantic, structural, and statistical matching techniques to align multiple ontologies using an agent-based multimodel, and 3) evaluate the effectiveness of the generation and alignment processes using precision, recall, and various other measures of effectiveness. Preliminary results of our initial agent-based experiments were promising – by applying ensembles of multiple matching techniques, we achieved significant improvements in alignment F-scores and other measures of performance while dramatically reducing the amount of time required to manually produce coordinated, useful domain models.**

Keywords-ontology; ontology alignment; social ontology; cultural ontology; ensemble alignment; agent-based alignment

I. INTRODUCTION

Having an incomplete understanding of the social-cultural factors that define and influence a region can cripple an otherwise promising business, political, or military strategy. Too often, models that guide strategy development and operational planning do not include critical social and cultural elements. These omissions can be blamed partly on oversight, but often stem from a fundamental lack of understanding of how to model such difficult and unfamiliar concepts. The information required to generate useful contextual models is typically available but is often distributed across vast repositories. Furthermore, the manual interpretation of detailed text is time-consuming, highly subjective, and requires specialized skills. We believe that socio-cultural awareness is best achieved by a system that combines multiple information sources using a variety of automated extraction, mediation, and analysis tools, but guided by a human knowledge engineer in an interactive paradigm called *balanced cooperative modeling* [1].

We apply *ontology* as our modeling method of choice. An ontology can conceptualize a complex domain in a way that both humans and machines can understand, but the use of ontology in this context presents us with two important challenges. First, manual ontology creation is a time-consuming and highly subjective process, particularly when attempting to model abstract social and cultural concepts. While formal models are required to conform to strict rules involving provable logic and model consistency, they will always incorporate some amount of bias. Every human modeler will have a slightly different perspective of the same small part of the world, and will make different value judgments about what parts are important and how those parts interrelate. Striving for added richness by adding more information only complicates this problem and adds to the severity of the "knowledge acquisition bottleneck" [2]. We believe, therefore, that by applying automated *ontology generation* against various corpora of domain-relevant materials, we can generate a useful first approximation of a domain model. An automated generation process will "learn" from the information it can "study". The model it constructs will, therefore, be representative of the "world" described in the input material it receives.

The second challenge involves the alignment of multiple models. Accommodating multiple domain ontologies is usually necessary to capture the complexities of domains having socio-cultural dimensions and to leverage existing models. There has been a loosely-associated body of work in this area that we collected under the general heading of "*ontology alignment theory*". Our interpretation of this theory is essentially built on the principle of approximation – because any ontology is an approximate representation of its real-world domain, generating and aligning multiple ontologies that all represent the same domain yields a richer higher-order approximation of the real world (i.e., removes some of the subjectivity or bias associated with applying a single model).

As described by Euzenat and Shvaiko [3], the *matching operation* accepts ontologies as inputs, and produces an ontology as its output (see Figure 1). The input ontologies (O_1 and O_2) are independent domain ontologies, perhaps derived from different sources of information or developed by different ontology engineers. Optionally, a third ontology (Ω) may be included as input – this may be an upper ontology or may be the composite ontology (or alignment) produced by a

The authors are grateful to have received funding for SCUBA under the 2010 Raytheon Innovation Challenge.

previous matching operation. The latter case suggests that matching operations can be chained for continued refinement by feeding the output from one operation (i.e., an aligned ontology) as input to the next matching process.

$$\Omega' = f(O_1, O_2, \Omega, p, r)$$

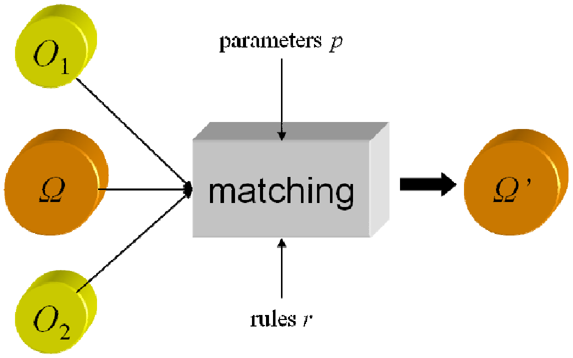


Figure 1. Matching operation, from Euzenat and Shvaiko [3]

In addition to the ontologies, a couple of additional inputs are provided. First, a set of rules directs the matcher to perform certain types of comparisons (i.e., which entities or attributes to compare, what sort of comparison to make, etc.). These rules are derived from a set of basic matching techniques described later in this paper. To accompany the rules, a set of parameters informs the matcher what limits or constraints to impose on the rules. For example, a rule might cause a name similarity technique to be performed using a fuzzy string comparison on a "name" property, but a parameter might indicate that only values having a confidence value higher than 50% are to be considered a match. Finally, the result of this operation is an ontology, referred to as *Ω prime*, that expresses the set of correspondences between the entities in O_1 and O_2.

Considering the points made above, we set two primary goals for our project: *Eliminate the knowledge acquisition bottleneck through semantic parsing and extraction of domain concepts from data sources into multiple ontologies and contexts,* and *bridge the gap between multiple, heterogeneous generated ontologies and a single domain ontology.* In our initial phase, we chose to apply the previous work of others in generating ontologies from text using readily available tools (citations to follow). Our investigation, instead, focused on the effective alignment of ontologies through various techniques of mapping. Here, too, we borrowed from the work of others for specific techniques and algorithms (citations to follow). However, we began with the premise that ontologies have characteristics that make them more or less suited for effective alignment with certain other ontologies. Rather than approach the problem using a single technique or by applying complex n-way comparisons, we formulated three key observations that guided our efforts:

Observation #1: Certain pairs of ontologies are more effectively aligned with one another than with other ontologies

Observation #2: Certain matching techniques produce more useful alignments for certain ontology pairs than others

Observation #3: The selection of candidate ontology pairings and matching techniques can be guided by heuristics and aided by the inspection of model metacharacteristics

This paper presents SCUBA, an agent-based framework for ontology alignment based on the observations stated above. We will describe the methodology we applied, as well as provide some initial results.

II. Methodology

The objective of SCUBA was to develop a balanced human-computer modeling paradigm to 1) automate the creation of social and cultural ontologies from selected source materials, 2) apply a variety of nominal, semantic, structural, and statistical matching techniques to align multiple ontologies, and 3) evaluate the effectiveness of the generation and alignment processes. Since our work was mainly focused on the alignment framework, we will concentrate most of this section on that effort.

A. Ontology Generation

In answering the challenge of ontology generation, we relied on the groundbreaking work of a number of others, particularly Maedche and Volz[4] and Cimiano and Völker [5]. We used the common academic ontology generator Text2Onto [6] to generate ontologies from bodies of text we obtained from various sources, including the Yale University Human Relations Area Files (HRAF) [7], Yale University's Outline of Cultural Materials (OCM) [8], the United Nations Development Programme – Human Development Reports [9] and others. Documents were clustered by geographic area, and a separate ontology was generated for each area. The team also generated 95 separate ontologies utilizing the 54 Cultural and 41 Social text files obtained from open source materials. All ontologies were created in the Web Ontology Language (OWL) format.

Additionally, the team manually generated a set of "gold standard" ontologies to compare with the automatically generated models. Seven ontologies were created based on the Department of Defense (DoD) PMESII-PT paradigm (Political, Military, Economic, Social, Information, Infrastructure, Physical Environment, Time) [10] using Protégé [11]. An ontology for Time was not created. To provide instance data for the PMESII-P ontologies, the team developed a method to automate the merging of Yale HRAF instance data with Yale OCM codes in the developed ontologies, saving weeks of manual labor.

B. Ontology Alignment

1) General Approach

As stated, we focused most of our work on ontology alignment. We again borrowed heavily from the body of prior research in specific ontology matching techniques, most of which were collected and documented by Euzenat and Shvaiko [3]. In order to investigate our own hypotheses, however, we constructed a customized agent-based framework using the Java Agent DEvelopment Framework (JADE) [12]. We used agents to develop automated workflows for the two main component processes: selecting the optimal set of alignment candidates and most promising match techniques, and performing the matching operation by applying the rules to the alignment candidates (Figure 2). Specific match techniques were encoded as composable sets of agent behaviors. An agent-oriented design allowed us to apply a technique known as "ensemble forecasting", which is common in highly specialized domains such as weather prediction. Yilmaz [14] refers to this idea as a *multimodel*, or a set of component models that, together, define the behavior of a more complex process. Using ensemble forecasting or multimodeling, various combinations of matching algorithms ("behaviors") were applied against concept pairs, then evaluated in order to determine the strength of the match. An average, or *ensemble mean*, of the different behaviors inspired greater confidence because it essentially smoothed the performance peaks and troughs introduced by model imperfections or context sensitivities. For example, the concepts "car" and "automobile" produce very low results for all name-based match behaviors, but semantic match behaviors rate them as nearly identical. Hence, while any one technique for matching two concepts is inherently unreliable, an ensemble mean that accounts for the strengths and weaknesses of all match techniques yielded a higher-confidence correlation. The matches can be used to produce a merged ontology in any format desired; e.g., a set of OWL assertions (i.e., "sameAs" or equivalentClass") between matched concepts, or Semantic Web Rule Language (SWRL) rules to bridge the aligned models.

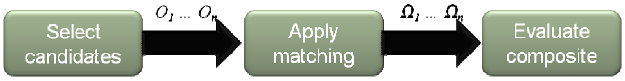


Figure 2. Primary SCUBA workflows

2) *Agents, Behaviors, and Ensembles*

In the SCUBA framework, a community of agents interacts to perform the high-level operations of candidate selection and ontology matching. Each agent determines the types of behaviors it needs in order to perform its current task, and loads them dynamically. Agents serve in a variety of roles:

- OA - *Ontology Agent*: perform as a proxy for an ontology by mediating access to its concepts as well as responding to inquiries about its metacharacteristics (e.g., depth, breadth, number of concepts, etc.).
- EA - *Evaluation Agent*: make a judgment as to the relatedness of available ontologies along some relevant dimension (e.g., domain relevance, semantic similarity, etc.).
- HA - *Heuristic Agent*: determine which ontology pairs make good candidates for matching, which matching behaviors should be applied, and manage the execution of selection and matching workflows.
- MA - *Matching Agent*: creates mappings of the concepts and relationship types between two ontologies.
- SA - *Similarity Agent*: calculates the similarity between concepts.
- UA - *Utility Agent*: performs supporting tasks such as data and ontology storage/retrieval, job ID management, etc.

Each matching algorithm or technique was implemented as a behavior. In JADE parlance, a behavior is a set of actions to be performed. Coding each set of actions in a separate component, rather than in the agent itself, allowed each agent to select and compose the behaviors it wished to use to complete an assigned task. There are many techniques for performing the matching operation, and some are better at matching certain types of entities and properties than others. Furthermore, a better match might result in some cases if more than one technique is applied at the same time ("matcher composition").

- Name-based ("terminological") techniques compute some measure of similarity based on strings containing names, descriptions, comments, etc. Comparisons based on simple or fuzzy string comparisons would match "George Bush" with "George Bush", "George W. Bush", or "G. W. Bush". Matching can also be performed using synonyms ("newspaper" matches "periodical") or other language-based methods like lemmatization, which would match houses to house, mice to mouse, etc.
- Semantic techniques rely on deductive methods to justify their matching results. A semantic model could contain a very rich set of relations, with inferred associations between ontologies. For example, "brain injury" and "head injury" might be inferred to be synonymous based on the fact that a "brain" is "part-of" a "head".
- Structural techniques take into account an entity's attributes or properties, as well as other related entities, when performing a match. For example, a constraint-based rule would match "Book" and "Volume" if each contained the key properties of author, year, publisher, and title. Similarly, a graph-based rule would match "Book" and "Volume" if the two concepts had the same (or similar) subclasses, like "Novel", "Textbook", and "Children's".
- Extensional techniques are applied not to concepts, but to instances. Typically applied when other techniques contain little name or structure overlap, these techniques entail matching two concepts based on their membership; i.e., the objects that belong to

each particular class. For example, book titles are unique enough that, with some estimable probability, two instances having same title or label are likely to be the same object. If the object is classified differently in two separate ontologies, a match between concepts then becomes possible.

3) *Workflow Heuristics*

Heuristics are encoded inside an Heuristic Agent and govern the selection and matching workflows. Many such heuristics can be encoded simultaneously in one or many agents, and a single HA can construct complex heuristic workflows from multiple matching behaviors chosen from different categories. For instructional purposes, the following example is used throughout the rest of this section to describe what happens in each step of the process:

Agent: HA01
Behavior: MinDepth-MaxDepth
Other agents: EA01, OA1-OA_n, MA01, SA01-SA03
Behavior Description: Inspect each ontology for its depth. As candidates, choose the ontology with the minimum depth to be matched with the ontology having the maximum depth. Perform an alignment of the two candidates using an average of all available name-based and semantic matching techniques. Evaluate the results using an F-score statistic.

In the example, HA01 performs candidate selection by directing EA01 to evaluate ontologies according to their depth. EA01 requests a depth statistic from each of the OAs, and reports the results – the ontologies having the least and greatest depths – back to HA01. The candidate ontologies have now been identified, and the first phase is complete. HA01 then moves into the ontology matching phase. The agent directs MA01 to match the selected ontologies using all of the name-based and semantic behaviors. MA01 manages the next level of orchestration, directing a set of SAs to perform an alignment, assigning each to use one of the specific matching behaviors. For example, if there are defined behaviors for Levenschtein distance (name-based), Jaro-Winkler (name-based), and WordNet similarity (semantic), the MA tasks three SAs – one per behavior - to align the concepts in the candidate ontologies and record the results of their work. MA01 then computes the ensemble mean and reports its result back to HA01. Once the match process is complete, other components can refer to the scores in order to produce a number of possible outcomes: a merged ontology, a set of rules mapping pairs of similar concepts, ontology entries reflecting class equivalencies, etc.

4) *Workflow 1: Candidate Selection*

All available ontologies are evaluated and compared according to a subset of predetermined set of criteria (e.g., depth, breadth, domain relevance, number of concepts, etc.). From this observation, the most suitable pairs are selected for alignment. Additionally, matching techniques are chosen to maximize the effectiveness of the alignment process for the types of ontologies chosen as candidates. Figure 3 describes the roles of agents and behaviors in the candidate selection process.

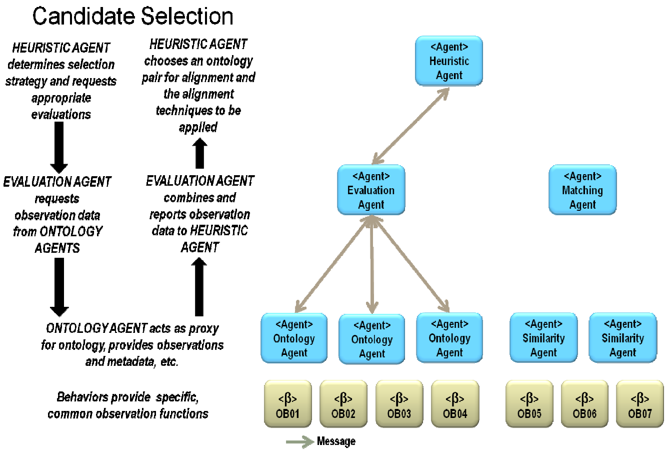

Candidate Selection
HEURISTIC AGENT determines selection strategy and requests appropriate evaluations
HEURISTIC AGENT chooses an ontology pair for alignment and the alignment techniques to be applied
EVALUATION AGENT requests observation data from ONTOLOGY AGENTS
EVALUATION AGENT combines and reports observation data to HEURISTIC AGENT
ONTOLOGY AGENT acts as proxy for ontology, provides observations and metadata, etc.
Behaviors provide specific, common observation functions
<Agent> Heuristic Agent
<Agent> Evaluation Agent
<Agent> Matching Agent
<Agent> Ontology Agent
<Agent> Ontology Agent
<Agent> Ontology Agent
<Agent> Similarity Agent
<Agent> Similarity Agent
<β> OB01
<β> OB02
<β> OB03
<β> OB04
<β> OB05
<β> OB06
<β> OB07
Message

Figure 3. Candidate selection workflow

5) *Workflow 2: Ontology Matching*

Once the candidate ontologies and techniques are chosen for alignment, the matching process is carried out using the agents and behaviors described above. Figure 4 illustrates the ontology matching process.

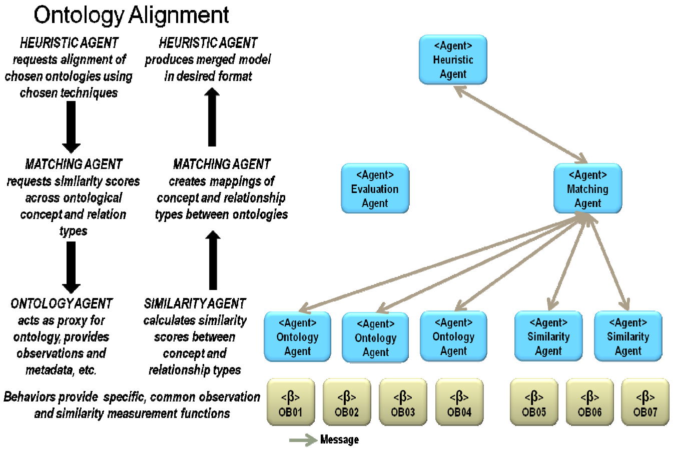

Ontology Alignment
HEURISTIC AGENT requests alignment of chosen ontologies using chosen techniques
HEURISTIC AGENT produces merged model in desired format
MATCHING AGENT requests similarity scores across ontological concept and relation types
MATCHING AGENT creates mappings of concept and relationship types between ontologies
ONTOLOGY AGENT acts as proxy for ontology, provides observations and metadata, etc.
SIMILARITY AGENT calculates similarity scores between concept and relationship types
Behaviors provide specific, common observation and similarity measurement functions
<Agent> Heuristic Agent
<Agent> Evaluation Agent
<Agent> Matching Agent
<Agent> Ontology Agent
<Agent> Ontology Agent
<Agent> Ontology Agent
<Agent> Similarity Agent
<Agent> Similarity Agent
<β> OB01
<β> OB02
<β> OB03
<β> OB04
<β> OB05
<β> OB06
<β> OB07
Message

Figure 4. Ontology alignment workflow

6) *Scoring and Evaluating*

All of the concept matching results were recorded in a database for later use. The entries included the two concepts being matched and their ontologies of origin, the behavior used to perform the match, and the similarity score that was normalized to range between -1.0 (known to be different) and 1.0 (known to be the same). A score of 0.0 indicated complete uncertainty. The scoring results were used to compute the ensemble mean over each discrete concept match (i.e., by averaging the scores of all behaviors that were applied to each of the match pairs).

C. *Demonstration*

A military planning exercise was chosen as a scenario to demonstrate SCUBA, since this type of event is typically a time consuming, manual, and ad hoc process that can take hours to days depending on size of the mission and echelon of command. War planners skim through available classified sources of information such as Signal Intelligence (SIGINT), Communications Intelligence (COMINT), and Human Intelligence (HUMINT), but typically spend the majority of their effort analyzing Open Source (OSINT) or simply searching the Internet[1]. As a result, critical information and cross relationships between pieces of information are commonly missed due to time constraints and the limits of human processing ability. Compounding the difficulty of the research effort, the number of data sources is necessarily limited by time and staff and not all information may be up-to-date.

1) *Military Planning Scenario & Decision Making Model*

The Military Decision Making Process (MDMP) Model [13] is a standardized mission planning and decision making model used by the US Army and combatant commands (COCOMs) to support counterinsurgency operations (COIN). The formal tactical planning process of counterinsurgency operations is performed by the commander's staff utilizing the MDMP model. In plain language, MDMP identifies the problem, develops solutions, compares alternatives, and recommends a best decision to the commander.

a) *Mission Analysis*

Mission analysis is crucial to the MDMP. It allows the commander to begin the battlefield visualization. The outcome of mission analysis is a tactical problem definition that feeds the process of determining feasible solutions. Mission Analysis consists of 17 steps, not necessarily sequential, and results in a formal staff briefing to the commander. Figure 5 depicts the breakdown of the MDMP model and green shading is used to highlight the relevant steps for the SCUBA demonstration.

b) *Initial Intelligence Preparation of the Battlefield (IPB)*

IPB is a systematic, continuous process of analyzing the threat and the effects of the environment on the unit. It identifies facts and assumptions that determine likely threat COAs. The IPB supports the commander and staff and is essential to estimates and decision making. It provides the basis for intelligence collection and synchronization to support COA development and analysis. Furthermore, it is a dynamic process that continually integrates new intelligence information.

IPB defines the battlefield or operational environment in order to identify the characteristics of the environment that influence friendly and threat operations, help determine the area of interest, and identify gaps in current intelligence. IPB describes the battlefield's effects, including the evaluation of all aspects of the environment with which both sides must contend, to include terrain and weather and any infrastructure and demographics in the area of operations (AO). IPB evaluates the threat by analyzing current intelligence to determine how the threat normally organizes for combat and conducts operations under similar circumstances.

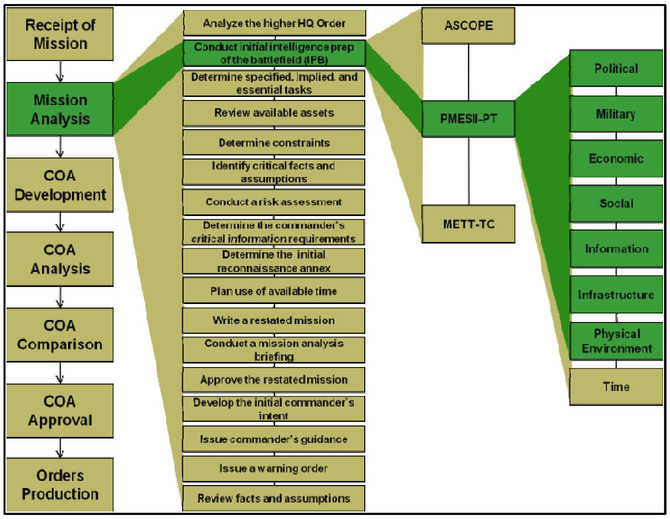

Receipt of Mission
Mission Analysis
COA Development
COA Analysis
COA Comparison
COA Approval
Orders Production
Analyze the higher HQ Order
Conduct initial intelligence prep of the battlefield (IPB)
Determine specified, implied, and essential tasks
Review available assets
Determine constraints
Identify critical facts and assumptions
Conduct a risk assessment
Determine the commander's critical information requirements
Determine the initial reconnaissance annex
Plan use of available time
Write a restated mission
Conduct a mission analysis briefing
Approve the restated mission
Develop the initial commander's intent
Issue commander's guidance
Issue a warning order
Review facts and assumptions
ASCOPE
PMESII-PT
METT-TC
Political
Military
Economic
Social
Information
Infrastructure
Physical Environment
Time

Figure 5. Military Decision Making Process Model

c) *PMESII Ontology Structure*

PMESII-PT, or for brevity PMESII, is a framework used to describe and understand the operating environment [10]. PMESII provides structure to the IPB process, and facilitates the organization of facts and assumptions about actors operating in an AO. Each letter in the PMESII acronym corresponds to a specific variable of interest to the war planner: P – Political, M – Military, E – Economic, S – Social, I – Information, I – Infrastructure, P – Physical Environment, and T – Time.

2) *Military Planning Using SCUBA*

As illustrated in Figure 5, the eighth variable, Time, was not modeled in this scenario since the time element was already embedded in the instance data populating the other variables. The DoD currently uses an expansion of the PMESII model that includes about 60 sub-categories. By merging PMESII with the Yale OCM model the SCUBA team extended the level of fidelity to approximately 900 super class and class concepts providing much greater model fidelity. An example of this expansion for the Social PMESII variable can be seen in blue in Figure 6.

When SCUBA executes, it ingests data from text documents, extracts domain relevant concepts, and links those concepts both vertically within individual PMESII variables and horizontally across the PMESII model. Instance (source) data is connected to each concept, which allows later review by the war planner or intelligence analyst. The main advantage of this paradigm is that instead of a planning staff performing manual keyword search queries across a variety of databases, a single lookup within SCUBA will provide the analyst or operational planner with all relevant information on a historic, social, or cultural topic of interest.

[1] This process was described to the SCUBA team during a December 2010 visit to the Joint Operations Center at US Central Command Headquarters, MacDill AFB, FL.

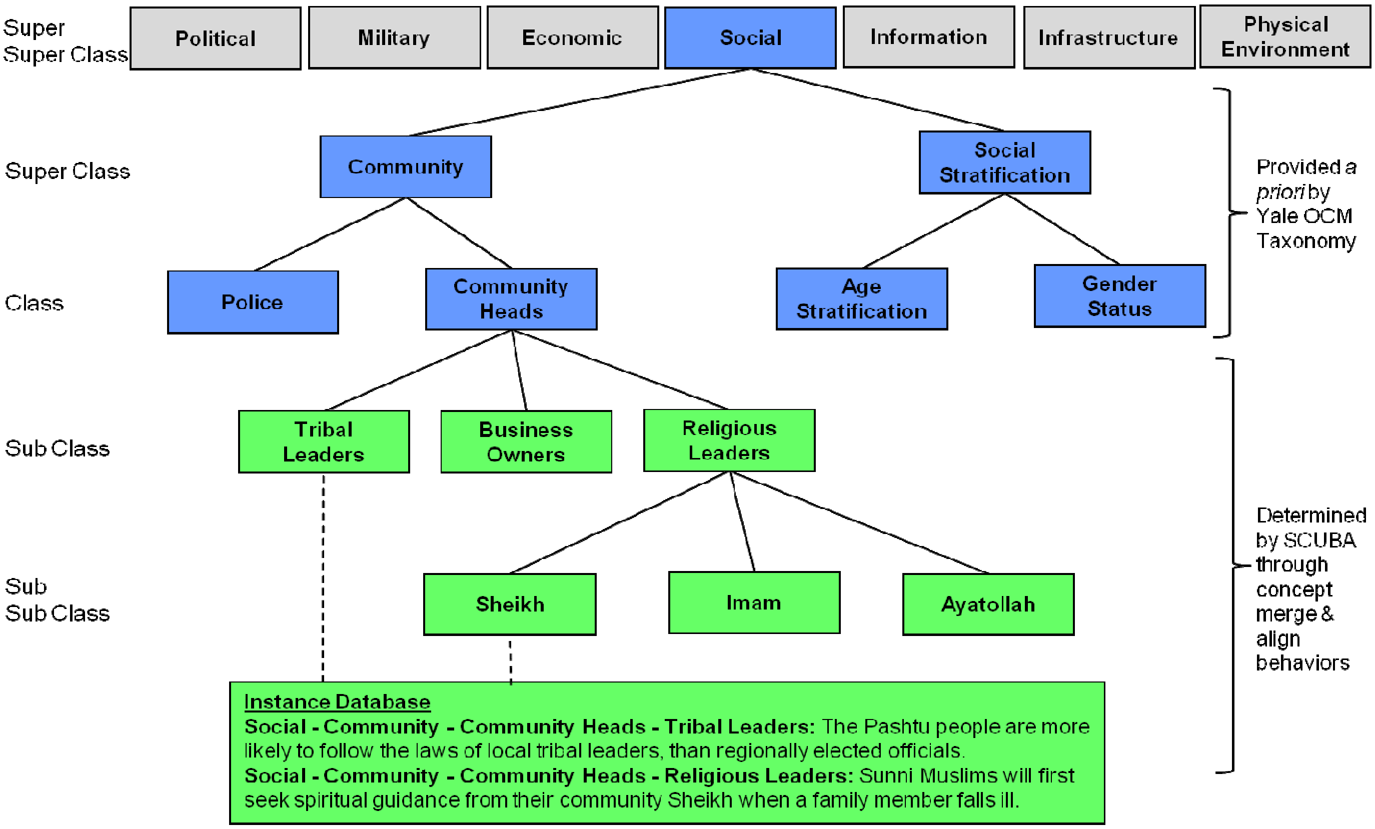

Super
Super Class
Political
Military
Economic
Social
Information
Infrastructure
Physical Environment
Super Class
Community
Social Stratification
Provided a *priori* by Yale OCM Taxonomy
Class
Police
Community Heads
Age Stratification
Gender Status
Sub Class
Tribal Leaders
Business Owners
Religious Leaders
Sub
Sub Class
Sheikh
Imam
Ayatollah
Determined by SCUBA through concept merge & align behaviors
Instance Database
Social - Community - Community Heads - Tribal Leaders: The Pashtu people are more likely to follow the laws of local tribal leaders, than regionally elected officials.
Social - Community - Community Heads - Religious Leaders: Sunni Muslims will first seek spiritual guidance from their community Sheikh when a family member falls ill.

Figure 6. PMESII ontology structure

While the focus of SCUBA is in the socio-cultural domain, an expanded PMESII model was created in order to demonstrate horizontal relationships between variables. The result is a major improvement over existing systems that are highly specialized and restricted in scope. Additionally, when this same effort is performed manually, the PMESII variables are commonly divided between staff officers. This practice produces information stovepipes and complicates the task of identifying cross-variable effects. In contrast, SCUBA facilitates such understanding.

Continuing with the example in Figure 6, the concept class hierarchy in green are those identified and created by SCUBA. Notice that in addition to aligning similar concepts, SCUBA can create class hierarchies, merge similar concepts into a single class, and link original instance data to each relevant concept.

A small portion of the merged PMESII Ontology generated by SCUBA using open source socio-cultural information of Afghanistan was displayed in Raytheon's hyperbolic semantic graph tool and is shown in Figure 7. Notice the equivalence relationship identified between "Military Organization" and "Militia". Also, "Districts" in one ontology was aligned with "District" (no 's') in another ontology. This was all performed automatically by the SCUBA agents and behaviors with no human in-the-loop. In the case of Districts/District, the SCUBA heuristic relied on structural matching techniques. The match occurring between Military Organization and Militia was a combination of structural and semantic matching. The remainder of the figure illustrates multiple concepts arising from a single paragraph: Military Organization, Districts, and Police, as well as, additional instance data on each of those concepts arising from other source material.

III. Results

The team identified dozens of possible evaluation metrics, many of which were used in the candidate selection process. As an overall measure of effectiveness, however, we report our results in terms of F-scores using the formulas below.

$$F = 2 * \frac{precision * recall}{precision + recall} \qquad (1)$$

$$precision = \frac{tp}{tp + fp} \qquad (2)$$

$$recall = \frac{tp}{tp + fn} \qquad (3)$$

The F-score is a measure of a test's accuracy which considers both the precision ("exactness") and recall ("completeness") of the test. In the models we chose for testing, over 60,000 comparisons were made between

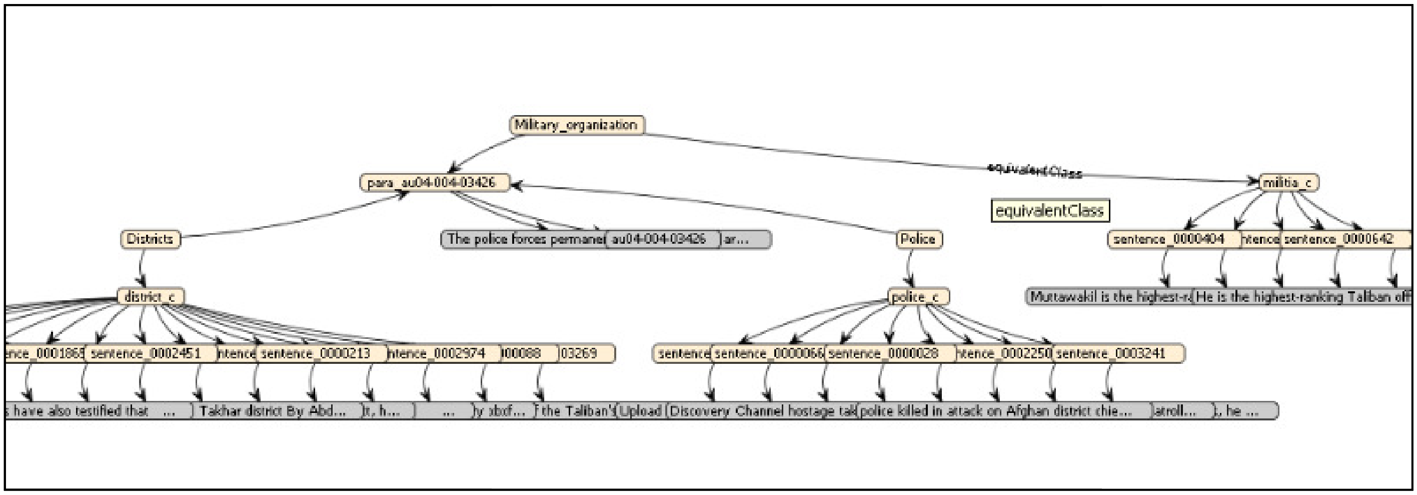


Figure 7. PMESII ontology structure in hyperbolic browser

concepts. Using name-based comparison alone, precision was typically high (~90%), while recall was much lower (~20 - 30%). Because the name-based approach suffered from a high number of false negatives, the F-scores averaged only ~40% (see Figure 8). However, when semantic matching was combined with name-based matching, there was a dramatic reduction of false negatives - this resulted in significant increase in recall (~80%) and brought the average F-score to above 80% (all differences were significant) – see Figure 9. Even greater improvement is expected when additional behaviors are added. Based on these results, we are encouraged by the prospect of evolving information alignment and interoperability from a manual, costly chore to an effective semi-automated process.

A. *Measures of Performance*

In order to determine whether the automated align-and-merge methodology defined by SCUBA demonstrated any improvement over existing ontology generation tools alone, the SCUBA merged ontology was compared against an ontology generated using Text2Onto [5][6]. In both cases, the same data set and initial taxonomy were used. The comparison was made across 12 measures of performance (MOPs) that fall within three general measurement dimensions: Structural, Usability, and Timeliness.

Structural MOPs

- Measure of Concept Count – Total number of concepts in the ontology.
- Measure of Concept Instance Count – Number of linked paragraph instances over all concepts.
- Measure of Relationship Type Count – Total number of unique relationships in ontology, i.e. 'is a part of', 'is equivalent to', etc.
- Measure of Relation Instance Count – Number of relationship links between concepts.
- Measure of Maximum Depth – Levels of concept hierarchy within the ontology.
- Measure of Degree Centrality - Measure used often in social network theory - average number of relationships linked to each concept.

Usability MOPs

- User Recognition – Survey score indicating how similar ontology structure is with current models.
- Fitness for User – Survey score indicating how easy it is for the user to load and navigate among the concepts in the ontology

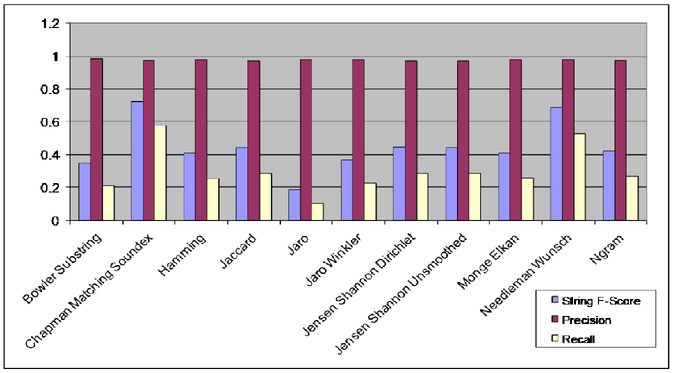


Figure 8. Results for string-only alignment

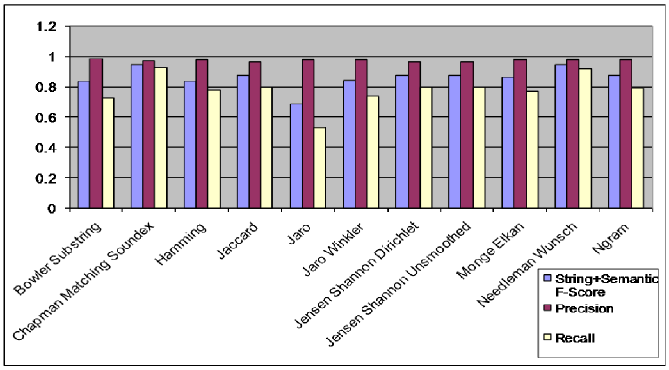


Figure 9. Results for String + Semantic Alignment

Timeliness MOPs

- Speed to Build Ontology – Time to create the ontologies.
- Time to Perform Alignment – Time to perform alignment between 2 or more ontologies.

Figure 10 depicts these MOPs for each of the ontologies. It is immediately clear that the Human Generated ontology was most recognizable with information in a format most easily used while the purely machine generated, Text2Onto ontology scored lowest in this area. SCUBA scored well in this area because it was based on the same PMESII model used by military planners. Conversely, the Human Generated ontology took longest to build and was much smaller than the faster and larger generated Text2Onto ontology. These results were in line with our expectations. What we intended to see was whether SCUBA could create ontologies that were at least as large/deep as those created by software algorithms or humans, but were richer and more usable similar to those generated by humans.

When breaking down the Speed to Build by individual algorithms, the SCUBA string-based matching agents and ontology behaviors were executed on par with Text2Onto, while the semantic matching agents took considerably more time to execute. This is understandable because the semantic algorithms are more complex with the purpose of determining additional positive matches through synonym, lemmatization, and morphological comparisons. This significantly improved the accuracy of the results, as boldly illustrated in Figure 9, however there is a corresponding increase in ontology generation run time. We believe this is reasonable (it is still significantly lower than the Human Generated ontology) and can be further reduced by adding computing resources.

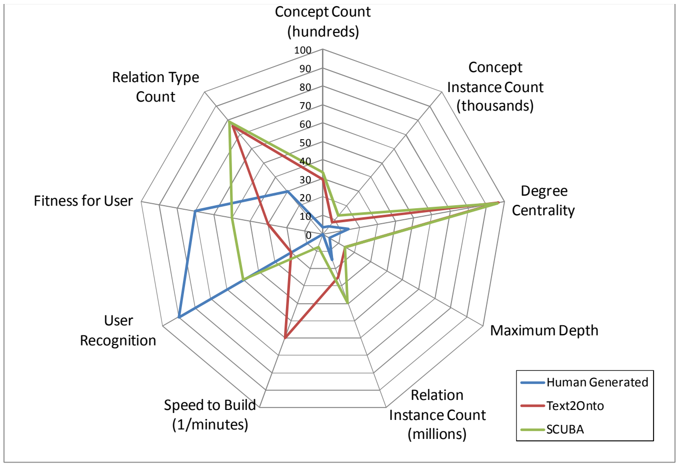


Figure 10. MOP comparison between Text2Onto and SCUBA

Regarding the other MOPs, SCUBA either met or exceeded the performance of Text2Onto. For example, for Concept Instance Count and Relation Instance Count, SCUBA identified close to 50% more concept and relationship instances than Text2Onto. This is an indication that the strategy of generating multiple smaller ontologies, and then aligning and merging the results into a larger composite ontology can improve information quality. Thus, SCUBA seems ideally suited for cases where information is spread across numerous and small data sources, or in cases where narrowly specific ontologies are merged with broader, more general ones.

IV. Summary

In this paper, we presented SCUBA, an agent-based ontology creation and alignment framework developed to address the shortcomings of current socio-cultural modeling efforts. SCUBA achieves a balanced human-computer modeling paradigm to 1) automate the creation of social and cultural ontologies from selected source materials, 2) apply a variety of nominal, semantic, structural, and statistical matching techniques to align multiple ontologies in a multimodeling environment, and 3) evaluate the effectiveness of the generation and alignment processes. Preliminary results of our initial agent-based experiments were promising – by applying ensembles of multiple matching techniques, we achieved significant improvements in alignment F-scores and other evaluation measures while dramatically reducing the amount of time required to manually produce coordinated, useful domain models.

Acknowledgment

The authors acknowledge the valuable contributions of the SCUBA development team at Raytheon IIS (State College, PA and Garland, TX) and Raytheon NCS (Largo, FL).

References

[1] K. Morik. Balanced cooperative modeling. *Machine Learning*, 11:217–235, 1993.

[2] B. G. Buchanan and D. C. Wilkins (editors). *Readings in Knowledge Acquisition and Learning: Automating the Construction and Improvement of Expert Systems.* Morgan Kaufmann, San Mateo, CA., 1993.

[3] J. Euzenat and P. Shvaiko, *Ontology Matching*. Springer-Verlag, Berlin Heidelberg (DE), 2007.

[4] A. Maedche and R. Volz, "The ontology extraction maintenance framework Text-To-Onto," in Proc. ICDM'01 Workshop on Integrating Data Mining and Knowledge Management, 2001.

[5] P. Cimiano and J. Völker. Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery. In Andres Montoyo, Rafael Munoz, Elisabeth Metais, Proc. 10th Int. Conf. on Applicat. of Natural Language to Inform. Syst. (NLDB), volume 3513 of Lecture Notes in Computer Science, pp. 227-238. Springer, Alicante, Spain, June 2005.

[6] http://code.google.com/p/text2onto/

[7] http://www.yale.edu/hraf/

[8] http://www.yale.edu/hraf/Short_OCM_List_121503.pdf

[9] http://hdr.undp.org/en/

[10] Field Manual 2-0, Intelligence. 2010. HQ, Dept. of the Army

[11] http://protege.stanford.edu/

[12] http://jade.tilab.com/

[13] Field Manual 101-5: Staff Organization and Operations. Dept. of the Army.

[14] L. Yilmaz, A. Lim, S. Bowen, and T. "Requirements and Design Principles for Multisimulation with Multiresolution, Multistage Multimodels" in *Proceedings of the 2007 Winter Simulation Conference,* Washington, DC, 2007.

Semantic Policy Enforcement and Reconciliation for Information Exchange in XMPP

Brian Ulicny, Won Ng, Oleg Simakoff, Jakub Moskal
VIStology, Inc.
Framingham, MA USA
{bulicny, wng, osimakoff, jmoskal}@vistology.com

Mieczyslaw M. Kokar
Department of Electrical and Computer Engineering
Northeastern University
Boston, MA USA
m.kokar@neu.edu

Abstract— **Extensible Messaging and Presence Protocol (XMPP) is a popular open-standard protocol for instant messaging (IM) widely used in military and commercial applications. In military contexts, as in commercial settings, it is often necessary to regulate who may communicate with whom and how. The distributed nature of XMPP makes centralized information exchange policy enforcement impossible, however. We report on a technology we have developed, called PolVISor, in which we express information exchange policies in a natural language formalism (SBVR SE), automatically translate these policies into an executable rule language (BaseVISor rule language) and enforce and reconcile disparate policies among XMPP servers, each with its own policies, using semantic technologies.**

Keywords: XMPP; security policies; policy reconciliation; SBVR; ontologies; deontology; modality

I. Introduction

Policy authoring, representation and enforcement are essential components in security systems. As systems grow and collaboration becomes more ubiquitous (e.g. via grid computing, collaboration among coalitions), the set of security policies grows larger. This leads to potentially undetected policy conflicts and the need for automated or semi-automated policy reconciliation. Our work has resulted in PolVISor, which uses ontological reasoning to determine security policy compliance and provide policy reconciliation when possible. We demonstrated the necessity, feasibility and flexibility of PolVISor to constrain information sharing in an XMPP (Extensible Messaging and Presence Protocol) environment.

II. Security, Policies and Reconciliation

In this project we were concerned with the ability to use policies to ensure compliance during runtime as well as with the ability to do policy reconciliation. Policy compliance involves the run-time process of ensuring that all of the conditions defined by a policy hold true; a common example is the checking of credentials required before granting access to a document. In policy reconciliation, the goal is to take multiple polices and, e.g., generate a policy instance that simultaneously satisfies all of them; a typical example here is determining specific conditions under which a communication session can be established between nodes in a VPN where the ends of the connection are governed by different policies.

1.1 Semantic and Non-Semantic Representations of Policies

Policies can be implemented in a system via the hardware (e.g. this light will not turn on unless both of these switches are turned on); or in software. In software, a policy can be represented either syntactically or semantically. By a semantic representation, we mean a representation in which inferences can be made on the basis of a policy instance using a domain-generic inference engine. So, for example, a Windows Group Policy instance has a meaning that is clear to everyone who knows the semantics of the policy language. However, no generic reasoning engine can draw inferences from Windows Group Policy instances in their native format. The representation has no meaning to those engines.

A primary objective in our work is to develop the means by which operations governing policies can be handled automatically by a computer. For this reason it is important to be able to describe policies in a formal, declarative way that will permit them to be automatically processed by formal reasoning engines.

A formal reasoner or inference engine is a system capable of applying the formal axioms of a language to a body of data/facts/knowledge resulting in the derivation of additional inferable facts. A rule-based system, for example, may be used as a formal reasoner if it is provided with a set of axioms for the language in which the data/knowledge is represented. Such axiom sets are available for a number of ontology languages as discussed below.

An important principle employed by many systems including policy-based reasoners is the use of the closed world assumption (CWA), which permits systems to assume that everything that is known to be true of the "world" is available in the facts that have been provided about it; if a fact is not explicitly stated it is assumed to be false. The closed world defined by a set of facts can be thought of as a "context" in which reasoning is to occur. OWL-based systems, like PolVISor, do not adopt the CWA.

For reconciliation to be possible there should be an explicit separation of policies and mechanisms that use the policies, and the policies should be first-class objects within the security system. In this way, policies will be objects that can be represented, stored and manipulated by the security system. Moreover, in this way policies will have their own interpretation, or semantics. This has a very important impact on the accreditation process in that mechanisms can be accredited and then policies can be added dynamically.

1.2 The Policy Reconciliation Problem

Two systems or elements of a system may impose policies on certain operations. In this paper we define policy reconciliation as the determination of a policy that implicitly or explicitly satisfies both policies and governs the behavior of the interaction of the system(s). Provisioning policies, authorization policies and information exchange policies are all types of policies that may require reconciliation.

In this project, we have bounded the problem of policy reconciliation in several ways. First, we assume that all partners in the policy negotiation process are equals. Therefore, we have chosen not to incorporate policy deference mechanisms saying that if System 1 and System 2 have different policies, then one of the system's policies overrides the other. While such meta-policies are widespread in practice, they do not pose an interesting conceptual problem.

Secondly, we have not dealt with preferences among policies. Thus, a system might allow distinct set of actions A or B (distinguished by their participants, say, or by other parameter settings), but it would prefer one set to the other. We have not addressed this issue because it essentially involves a different kind of modal reasoning: reasoning that ranks some situations as more desirable than others, although each is permissible. This is the logic of "should" and "should not", as opposed to the logic of "may (not)" and "must (not)" as described in the section on our deontic ontology of actions below. The considerations involved in modal reasoning about 'should' involves a higher-order reasoning than the logic of 'may' and 'must', and we have not addressed this in this project. In particular, we have not addressed what might be called "consequentialist" policies, where a policy is preferred based on its outcome. For example, one might say, choose policy A or policy B based on which one allows the most (or fewest) users (perhaps meeting some other criteria) to access some set of files. Such a system would require some kind of modeling and simulation step to determine how many users have access, and thus determine the policy choice.

Finally, we have not concerned ourselves with situations in which the policies to be reconciled cannot be completely disclosed between the interested parties. There are undoubtedly situations in which the policies that govern some action are themselves proprietary and sensitive in that they reveal, with contextual information, proprietary information. For example, suppose a University had a policy in which admitted students could sign up for a campus bulletin board system. If prospective students learned about this policy, they could potentially find out who had been admitted to the university before the official announcement had been made by trying to register on the bulletin board. In such a case, the university might want to avoid making such a policy known to other users or systems in order not to disclose unwanted information. We have not focused on such situations of policy reconciliation where trust is an issue since trust management is beyond the scope of our current investigations.

1.2.1 Information Exchange Policies

In this paper, we examine enforcing and reconciling information exchange policies. Information exchange policies are important in military and intelligence situations, where cross-organizational collaboration is required but strict policies restrict who can communicate with whom and what information they can exchange. For example, a military coalition might allow members of different national forces to collaborate on some tasks within certain channels and with certain information, but not others. The same is true of financial services and health care industries, which both regulate information exchange. For example, in financial services, so-called Chinese Wall policies regulate communication between analysts and traders. In health care, privacy and confidentiality policies regulate what information can be shared between health care providers and patients. Information sharing between social networking sites and other sites is another current example, particularly where single sign-on schemes like OpenID (http://openid.net) are involved.

In the military and intelligence community, information exchange policies are labeled "Cross Domain Solutions": "Cross Domain Solutions (CDS) are controlled interfaces that provide the capability to access or transfer information across different security domains." [1] The eXtensible Markup Language (XML) Data Flow Configuration File (DFCF) format specification[2] was developed to provide a common format for defining, validating, and approving XML data flows for use in XML cross domain solutions. DCDF is specified syntactically in XML in terms of information sharing system endpoints, where a complete policy specifies, for each endpoint pair, what information can be sent from an endpoint, and what information may be received by an endpoint. Such comprehensive policies are difficult to set up, are likely to become obsolete as the contents of the endpoint systems change, and are not flexible. Finally, they are not reconciled, across all endpoints because one system cannot impose any limitations on another system, only on itself. However, they can be implicitly reconciled at run time when two endpoints try to exchange information.

III. Policy Languages

In our project, we use SBVR Structured English (SE) for authoring policies in an English-like formalism. SBVR SE policies are then automatically translated into BaseVISor Rule Language (BVR) for execution and policy reconciliation.

[1] Unified Cross Domain Management Office, What is a cross domain solution?, http://www.ucdmo.gov/faqs.html.

[2] XML Data Flow Configuration File Format Specification Version 1.2.11 19 December 2008 http://iase.disa.mil/cds/helpful_tools/dfcf-specification-1-2-11.pdf

1.2.2 SBVR Structured English

Semantic of Business Vocabulary and Business Rules (SBVR) [1] is an OMG standard introduced in 2008 that aims at a more natural format for expressing rules. Business rules are expressed in a subset of natural language that is readily understandable by business people, instead of at an implementation level, such as rules that are processable by a formal reasoning engine. The vocabulary represents the concepts used in the rules and can also express facts and relations between concepts (e.g. that Fido is a dog). The specification is based on first order modal logic and captures the semantics of implementation-independent business models. Figure 1 locates SBVR in the Business Model (also called the Computation-Independent Model) level in OMG's Model Driven Architecture (MDA) [2] and is meant to be translatable to a Platform-Independent Model (PIM) that describes the structure and behavior of the model, and subsequently to a Platform-Specific Model (PSM) that includes all the platform dependent information necessary for a developer to implement executable code, such as specific programming language packages. SBVR is mapped to the Meta-Object Facility (MOF) [3] metamodel – a useful feature for transformations of an SBVR model to other models.

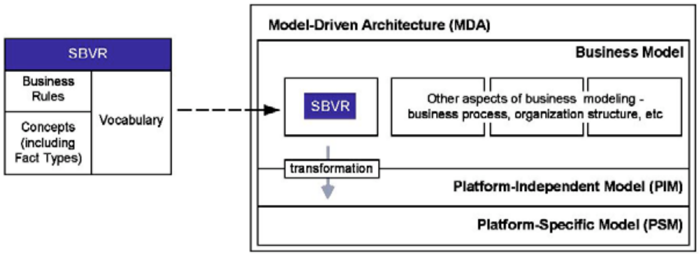

SBVR
Business Rules
Concepts (including Fact Types)
Vocabulary
Model-Driven Architecture (MDA)
Business Model
SBVR
Other aspects of business modeling - business process, organization structure, etc
transformation
Platform-Independent Model (PIM)
Platform-Specific Model (PSM)

Figure 1: SBVR in OMG's MDA.

SBVR distinguishes between *alethic* and *deontic* constraints. Alethic rules are categorized as structural business rules, which are rules that must necessarily be true as part of the business organization. Deontic rules are operative business rules that should be obeyed but which can be violated in practice.

SBVR has two common notations: Structured English and RuleSpeak®. Structured English (SBVR SE) is a controlled English vocabulary and grammar that uses font styling and color to indicate SBVR concepts. term represents a noun concept such as rule and action. Name is an individual concept and usually is a proper noun, e.g. California. *verb* is part of a SBVR construct called a fact type and is usually a verb, preposition or combination of preposition and verb. Lastly, SBVR SE defines a set of keywords that are reserved words or phrases with special meaning. Examples of keywords are the articles a and the, modality phrases It is necessary that, and quantifications every and at most one. An example of a SBVR SE rule is:

It is obligatory that a driver *is qualified* if the driver *rents* a car that *is owned by* EU-Rent

SBVR RuleSpeak® [4] is a proprietary variant developed by Business Rule Solutions, LLC (BRS) [5]. RuleSpeak® provides templates for business rules based on the category or subcategory that applies to the rule. We did not use the RuleSpeak format and will not address it here.

Like the other languages discussed, SBVR is domain and application independent. The SBVR specification includes a proposal relating SBVR concepts to equivalent OWL expressions, so clearly some consideration was given to how SBVR should work with semantic languages. Its main strength over the other languages is its user friendliness. Because SBVR SE is an almost-natural language, it is suitable for expressing high-level rules. Among available editors are SBVR-VE (SBVR Visual Editor) [7], a graphical drag-and-drop editor where attempts to create links between boxes containing, say, a modality and a term, would often not work; Sepiax-Web [8], an Ajax-based web editor with WordNet and SBVR integration; SBeaVer [9], an Eclipse plugin that provides syntax highlighting for SBVR SE; and a proposed SBVR tool component [10], including an editor, as part of Eclipse's Modeling Development Tools (MDT) [11] that has been in development for the past few years. There are also enterprise editors that support RuleSpeak®.

SBVR is sufficiently expressive for representing high level rules but because SBVR is at the business model level, it suffers from the common problem that most business model level components do: translation to a PIM and especially to a PSM requires additional details about computations and platform-specific information, usually supplied by an IT person. The SBVR vocabulary can be expanded to include platform vocabulary, but SBVR is meant to be a high level language and is not executable, so SBVR is most useful when translated into a lower level executable language like BaseVISor Rule Language (BVR), as we have done.

1.2.3 BaseVISor Rule Language (BVR)

BaseVISor (http://www.vistology.com/basevisor), a versatile forward-chaining rule engine specialized for handling facts in the form of RDF triples (i.e., subject, predicate, and object), expresses rules in BaseVISor Rule language (BVR). The BaseVISor engine implements OWL 2 RL inference rules in BVR and supports XML Schema Data Types.

Generally speaking, rules are expressed in the form of if/then statements. The 'if' part of the statement is represented by the 'body' or 'antecedent' of the rule; the 'then' part is represented by the 'head' or 'consequence'. In BVR the contents of rule heads and bodies are made up of triple patterns (i.e., triples that may contain variables) and procedural attachments, i.e. functions such as *add*, *assert*, and *println* (print line). Users can add user-defined procedural attachments for use in rules. BaseVISor also supports queries, which are special cases of rules with empty heads, and are useful for retrieving information from the resulting fact base.

BVR is domain and application independent, compatible with the semantic languages OWL and RDF, designed for formal reasoning and executable in the BaseVISor environment. It is very expressive, especially since the language is extensible via user-defined procedural attachments. A BVR editor is available as an Eclipse plugin to aid in composing BVR rules.

Translation of SBVR SE into BVR makes use of metamodels for both languages. First, SBVR SE expressions of policies are saved as XMI, then a proprietary metamodel-to-metamodel mapping is used to translate the SBVR XMI into a corresponding BVR rule, preserving its semantics.

IV. Security Policy Ontologies

We developed two OWL ontologies to encapsulate our treatment of policies as classes and to represent concepts and their relations that we have determined to be essential for security scenarios, including information exchange. These "core" ontologies are the basis for any domain-specific application of PolVISor, i.e. domain-specific scenarios should extend these ontologies with their domain-specific knowledge and rules. The design of the ontologies, such as treating actions and operations as first-class entities, are grounded in our study and investigation of formal security models.

1.2.4 Representing Modal Notions in OWL

PolVISor, as we have said, involves two kinds of modality, *deontic* and *alethic*. Modal expressions qualify the truth of a statement. For example, to say that "John is possibly dyslexic" is not to assert that "John is dyslexic", but a more qualified statement that the statement might be true. Modality is expressed logically as operators over propositions. Op(p) means that some modal operator *Op* is being asserted of the proposition p: *It is Op that p.* The operator identifies the way in which the truth of the bare proposition p is being qualified.

Alethic modality is the logic of possibility (it is possible that p) and necessity (it is necessary that p). As specified by SBVR, alethic notions are encoded directly in the ontology. Necessity relations between classes are expressed in terms of subclass relations that apply to all instances. Thus, to say that "necessarily, all bachelors are unmarried" or "necessarily, all cats are mammals" is to say that the class Bachelor is a subclass of Unmarried Things and that Cat is a subclass of Mammal. Without such a subclass relation, it might be that all of the instances of Bachelor are instances of Unmarried, but that would be a contingent coincidence, not a necessary truth, with respect to that ontology. We encode that it is possible that (some) Fs are Gs (e.g. that some File Clerks are Dyslexic) in the ontology by failing to have class F (File Clerk) and G (Dyslexic) as disjoint classes. If F and G are marked as disjoint classes, then necessarily, no Fs are Gs, (and, necessarily, no Gs are Fs), according to that ontology.

"It is necessary that a user has a password" expresses a necessity relation between the class of Users and the class of things that have a password. This necessity relation would be expressed by saying that the class of Users is a subclass of the class of things that have Passwords. This encodes the necessity relation in the ontology directly. Ontologies, after all, express constraints on how the world can be. To say that users *may* have a password is expressible by saying that the class of Users and the class of things that have a password are not disjoint.

Deontic Logic [12] is the study of the logic of the concepts "may" (or deontic 'can') and "must" and their duals "may not" and "must not". These concepts are crucial in expressing policies: policies express what may or may not be done, under certain conditions, and what must and must not be done, again under certain conditions. *May* and *must* are modal notions. Sentences employing modal notions do not express the way the actual world is, but qualify the truth of the proposition they modify, in this case expressing conditions on how possible worlds should be if they are to comply with the policies our ontology encodes. That is, if I say that "John may go to the store" or "John must (not) go to the store", I do not say anything about how the actual world is with respect to John's going to the store. What I express has to do with the consistency of John's going to the store with the ways in which John is permitted to act or with the ways in which John must act.

In our inference engine, BaseVISor, propositions are expressed as triples (subject, predicate, object). BaseVISor does not allow for modal operators over triples. Therefore, rather than give modal operators their usual semantics as quantifiers over possible worlds or ways the world could be or ways a person could act, we treat Actions as a class that can be subdivided into Permissible (may), Omissible (may not), Optional (may and may not), Obligatory (must) and Prohibited (must not) subclasses.

The structure of the ontology is represented in Figure 2:

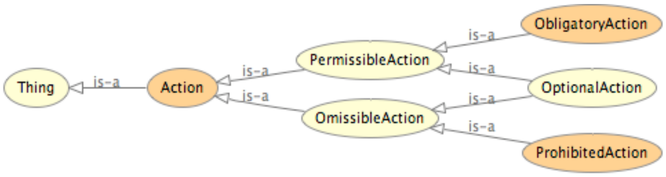


Figure 2. Classes and subclasses of Deontic Ontology

First, Actions are subclassified as Permissible or Omissible. An action is Permissible if it may be done. For example, getting married is permissible, so the class of actions that is getting married could be represented as a subset of the class of permissible actions.

An action is Omissible if it is permissible not to do it. For example, eating okra is omissible. One may abstain from eating okra. The class of actions that is okra-eating could thus be represented as a subset of the Omissible actions.

In fact, one both may and may not eat okra (and one may or may not get married), so instances of both of these types of actions would be instances of the intersection of the Omissible and Permissible classes: the Optional actions.

Obligatory actions (actions one must do) are a subset of the Permissible actions. If an action must be done, then it may be done. The Obligatory actions and the Omissible actions are disjoint: if an action must be done, it is not the case that it may not be done.

Similarly, Prohibited actions (actions one must not do) are a subset of the Omissible actions (actions one may not do). The Prohibited actions and the Permissible actions are disjoint: if an action must not be done, then it is not the case that it may be done.

We have expressed these relations in an OWL ontology. The ontology may be downloaded at http://vistology.com/ont/2010/secpol/Deontic.owl.

By means of this ontology, one can state that all instances of actions of a certain type are, for example, prohibited (e.g. theft, murder) or permissible (e.g. expressing one's opinion, forming associations) across the board. Policy rules allow one to express conditions under which actions of a certain type are classified as permissible or prohibited or optional based on additional facts about them. For example, one could express the policy that it is permissible to marry only if one is at least a certain age, not already currently married, and so on.

1.2.5 Upper Policy Ontology

We developed a policy ontology to serve as the base of all application- or domain-specific ontologies, available at http://vistology.com/ont/2010/secpol/UpperSecPolOnt.owl. It was derived by starting with the Naval Research Laboratory's (NRL) Security Ontology [25]. The NRL ontology was primarily designed for annotating resources with security-related metadata in order to facilitate the discovery of resources that meet security requirements.

In our ontology, a Policy consists of one or more Rules, associated with a SecurityPurpose (e.g. Data Integrity, Confidentiality). Rules are expressed in SBVR SE and translated into BaseVISor rule language. Rules govern Operations (Actions), i.e. operations performed by an element in the system (e.g. reading a file). Each Operation has an agent who originates the operation and an object that is the target of the operation. In the example of Bob reading a file foo.txt, the operation is a Reading with agent Bob and object foo.txt.

These Operations are declared to be owl:sameAs the class of Actions in the Deontic ontology, and thus, subclassified as Permissible and Omissible, and so on. They can also be equated to some other ontological representation of operations. Here we assert our Operation class to be owl:sameAs the class of UCore-SL Acts, indicated by the namespace sl. UCore-SL is an OWL version of the UCore [13][14] messaging format adopted for information sharing among the defense and intelligence communities.

For Security Markings, we have employed Richard Lee's ISM Ontology v. 0.7 [15]. This ontology is described as "a rendering of the IC-ISM XML spec for security markings. It is based on the IC-ISM v5 XSD, updated thru 2010-09-25. Although this ontology provides a complete taxonomy of security markings in use by US and Coalition partners, it does not generally order security markings from high to low within a markup scheme. We have added axioms to encode these facts as needed.

V. Information Sharing in XMPP

Extensible Messaging and Presence Protocol (XMPP) [16] is a popular open-standard protocol for instant messaging (IM) widely used in military applications. There are a number of extensions to the protocol that define protocols for other functionality, like Voice Over IP (VoIP). Each user signs into his XMPP account identified by a jid, commonly of the form name@domain.server, e.g. juliet@montague.net. Each jid has a contact list called a roster. Figure 3 illustrates the process when a user signs on. The server hosting the user automatically sends a presence to each of his contacts, except for those he has blocked, to indicate that he is now online. The contact's server forwards the presence to the receiver, unless she specified that she does not wish to receive presences from the sender. The contact's server also sends back a presence to the sender if she has not blocked presence-outs to the sender. Now the two clients can start chatting with each other. Users can also join chatrooms, participate in conversations as a group, and send messages to individuals in the room.

Privacy lists allow users to specify contacts with whom he wishes to restrict contact. However, there are currently no methods for server to specify policies to restrict users' chat, except by name. Using Openfire [17], an open source XMPP server available from Ignite Realtime [18], for our server, we developed an Openfire plugin that intercepts incoming and outgoing XMPP stanzas. The stanzas of interest in our scenarios are presences and messages, but all stanzas are intercepted so our implementation is extensible. Users connect to servers via Spark IM Client [19], an open source IM client application also provided by Ignite Realtime.

The Openfire plugin plays the role of the context handler here. It invokes an XSLT script to translate the XMPP stanzas to RDF and passes the RDF version of the stanza to PolVISor. PolVISor analyzes the stanza and returns to the plugin a decision to allow or deny the stanza. We chose to implement a deny-overrides approach, where if any applicable rule denies the stanza, the stanza is denied. If the stanza is allowed, the plugin forwards the original stanza to Openfire, which processes it as usual. If the stanza is denied, the plugin drops the stanza and the server does not see it. The behavior of the plugin can be changed to modify the stanza instead, for example if dropped stanzas should be logged by the server.

We developed an XMPP ontology with the core concepts such as jid and presence. The scenarios below build upon this base ontology. Because of our action-oriented approach in the upper ontology, actions like *Sends* are subclasses of Operation. BVR rules convert the stanza information to match the ontology, e.g. based on a presence stanza from juliet@montague.net to romeo@capulet.com, a *Sends* instance is generated that has agent the sender juliet@montague.net and has object the presence stanza, and the presence stanza has "to" romeo@capulet.com and "from" juliet@montague.net.

A. XMPP Presence Scenario

To demonstrate server policies that limit who can communicate with whom, based on facts about the persons involved, we implemented rules and ontologies for one server that restricts chat based on gender and another server that restricts chat based on the first letter of the jid. Gender is used for simplicity, but any class of persons could be used here, for example, filtering users by any combination of role or nationality or location. Because chatting depends on the initial sending of presences to contacts, the rules analyze presence stanzas and apply to incoming and outgoing presences. The rules state:

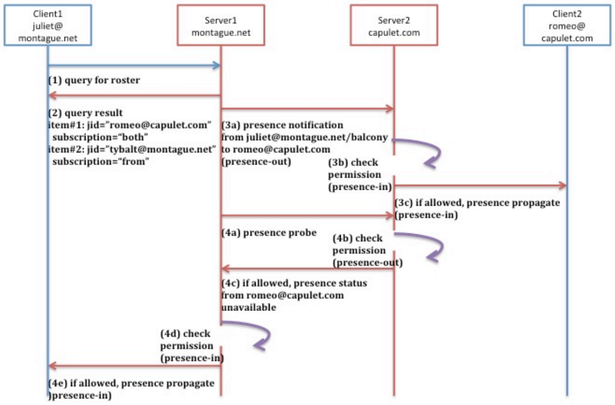


Figure 3: XMPP sequence diagram for sign in and roster retrieval.

Server 1 Policies:

Allow presences to/from Males on Mondays, Wednesdays and Fridays.

Allow presences to/from Females on Tuesdays, Thursdays and Saturdays.

Each user's gender is encoded using the FOAF (Friend of a Friend) vocabulary, and the information is available to Server 1. Because the FOAF gender is an untyped literal, a helper BVR rule determines whether a jid is an instance of the class Male or Female accordingly.

Server 2 Policies:

Allow presences to/from contacts whose jid start with A-L on Mondays, Tuesdays and Wednesdays.

Allow presences to/from contacts whose jid start with M-Z on Thursdays, Fridays and Saturdays.

Server 2's rules take advantage of BaseVISor's built-in regular expression procedural attachments.

All other stanzas that are not allowed explicitly are denied, e.g. no one hosted on either Server 1 or Server 2 can chat with others on Sundays. Here, policy reconciliation is implicit; a presence successfully sent from a user on Server 1 to a user on Server 2 means that both Server 1 and Server 2 allow the stanza. Therefore, amy@server1.com who is Female on Server 1 can chat with brenda@server2.com who is Female on Tuesdays because of Server 1's second rule and Server 2's first rule.

B. XMPP Security Labels Scenario

CWID 2010 featured a Cross Domain Collaboration implementation [20]. The collaboration scenarios included chatting and document sharing using security labels, access control and authentication. Clients and servers were modified to support security labels, among other functionalities. Boldon James's SAFE IM for XMPP [21] allows users to assign security labels to their one-to-one chats, group chats in rooms, and file transfers. It checks that receivers of labeled messages have sufficient clearance to read the message and that users who wish to join a chatroom with a security label have sufficient clearance to join. Isode's M-link server [22] is a XMPP server with support for controlling message flow based on the security label of the message and the security clearance of the sender and recipient.

We have implemented the same functionality of security labels by extending the ontologies and rules for the presence scenario outlined previously. Both sets of ontologies and rules for Server 1 and Server 2 have the same extensions and use the security levels from Intelligence Community Information Security Marking (IC-ISM) ontologies [23] for security labels and clearance levels. We added reflexivity and transitivity to the relevant properties so PolVISor can reason that someone with a clearance of TopSecret can send and receive messages classified as TopSecret or any lower level like Secret or Unclassified. This scenario considers one-to-one labeled chat messages but can be easily extended to group chat messages sent to a labeled chatroom so that no messages with a label at a higher level than the chatroom's maximum allowed label could be sent. Clients set the level by enclosing the label in brackets in the beginning of the message body, e.g. [RESTRICTED].

The rules state:

If a sender sends a labeled message to a recipient on a different server and the sender has equal or higher security clearance than the security level of the message, then the message is permitted to be sent.

If a recipient of a labeled message from another server has equal or higher security clearance than the security level of the message, the message is permitted.

If the sender and recipient of a labeled message are on the same server, and if the clearance of the sender and clearance of the recipient are equal or higher than the label's level, the message is allowed.

If a message does not explicitly have a security label, the message's security label is Unclassified.

All stanzas not explicitly allowed are denied.

C. XMPP Chatroom Reconciliation Scenario

To demonstrate explicit reconciliation, we implemented another scenario. If a client on Server 1 wants to join a chatroom hosted on Server 2 and both Server 1 and Server 2 have security policies restricting who can join what chatroom, then their policies must be successfully reconciled and the attempt to join must satisfy the reconciled policy in order for the attempt to be allowed. By satisfying the reconciled policy, the request also satisfies each server's policy. A client joins a chatroom by sending a presence to the chatroom, so the rules analyze presence stanzas.

The rules state:

Server 1 Policy: If the client is Male, he can join any chatroom.

Server 2 Policy: Any client can join any chatroom.

Server 1's policy is more restrictive than Server 2's, and lacking any other rules that concern clients joining chatrooms, ensure that only Males are allowed in chatrooms that involve any Server 1 clients. Figure 4 depicts the process. The plugin and PolVISor are not explicitly shown, but rather are subsumed as part of the server. The reconciled policy in this case is logically equivalent to Server 1's policy since Server 2's policy subsumes Server 1's. Therefore, reconciling the policies is equivalent to adopting the more restrictive policy.

However, because the servers have their own extended ontologies with server-specific classes and properties, ontology mapping could be needed for the request to satisfy the reconciled policy. An ontology mapping scenario is discussed below.

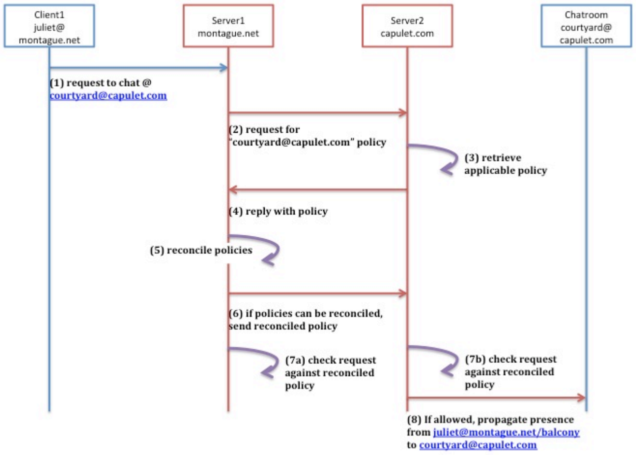

Client1
juliet@
montague.net
Server1
montague.net
Server2
capulet.com
Chatroom
courtyard@
capulet.com
(1) request to chat @ courtyard@capulet.com
(2) request for "courtyard@capulet.com" policy
(3) retrieve applicable policy
(4) reply with policy
(5) reconcile policies
(6) if policies can be reconciled, send reconciled policy
(7a) check request against reconciled policy
(7b) check request against reconciled policy
(8) If allowed, propagate presence from juliet@montague.net/balcony to courtyard@capulet.com

Figure 4: XMPP chatroom reconciliation sequence.

D. *XMPP Ontology Matching Scenario*

So far we assumed that both Server 1 and Sever 2 use the same ontology-based vocabulary to describe their clients. However, it is possible that the servers use facts expressed in different ontologies, in which case before polices can be reconciled, *ontology matching* must be first performed.

Ontology matching is the process of finding relationships between entities in two or more different ontologies. The output of matching, called an *alignment*, is a set of correspondences that express the relationship between different ontologies. Alignments include, but are not limited to, statements such as entity equivalence, sub-super relationship between entities, class intersection, or inverse relation. Alignments can be used to generate various tools used in further automated processing. For instance, a *translator* can translate data instances expressed in one ontology to another, or a mediator that can translate queries expressed in one ontology to another, and translate answers in the opposite direction.

Despite sophisticated methods from AI, ontology matching currently can rarely be fully automated beyond relatively simple correspondences, covering syntactic and terminological heterogeneity. When the same concepts in different ontologies are defined using different axioms, matching algorithms often have difficulties identifying any correspondences at all, or find ones that are irrelevant. When matching is incomplete or incorrect, manual editing is necessary. Matching systems typically allow the user to specify a threshold for confidence held in the correspondences, which allows for eliminating matches that are most likely invalid.

In order to demonstrate the use of automated ontology matching in the process of policy reconciliation, we matched FOAF and vCard ontologies with the threshold of 0.9 (1 being 100% confident) using an ontology matching API [24] and dynamically found the following relationships:

Equivalent classes:
- Foaf:Organization and vcard:Organization

Equivalent datatype properties:
- foaf:givenname and vcard:given-name
- foaf:givenName and vcard:given-name
- foaf:nick and vcard:nickname
- foaf:title and vcard:title
- foaf:family_name and vcard:family-name
- foaf:familyName and vcard:family-name

Equivalent object properties:
- foaf:logo and vcard:logo

Since vCard does not include gender information, we could not directly use this property to define policies. Instead, we used the nickname information, supported by both ontologies, in order to encode the gender of a user. We assumed that all nicknames follow a pattern "g_Name", where "g" can be either "F" or "M", indicating the user's gender. We designed a scenario similar to the XMPP Chatroom Reconciliation Scenario, with the following rules:

Server 1 Policy: If the client is Male (i.e. if his foaf:nick starts with M_), he can join any chatroom.

Server 2 Policy: Any client can join the chatroom.

Reconciled Policy: If the client is Male (i.e. if his foaf:nick or vcard: nickname starts with M_), he can join any chatroom.

While the policies are similar to the previous scenarios, this time Server 1 defines its policy using the FOAF vocabulary and imports the client's FOAF file, while Server 2 encodes user information in the vCard vocabulary. Thus, before the policies can be reconciled, FOAF and vCard need to be first matched in order to produce bridge axioms. Once matched, PolVISor reconciles the two policies, resulting in the reconciled policy equivalent to that of Server 1. Figure 5 shows the process.

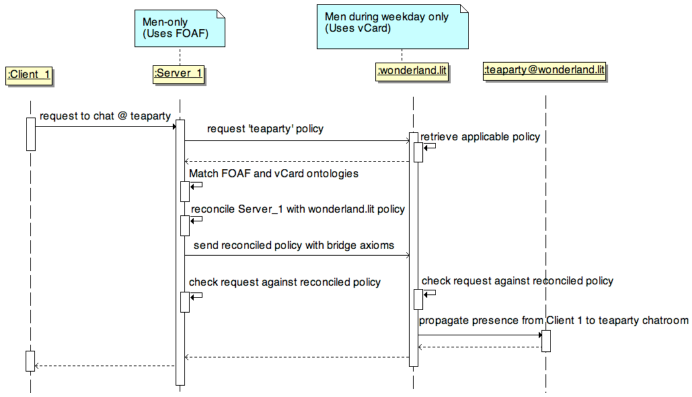

Men-only
(Uses FOAF)
Men during weekday only
(Uses vCard)
:Client 1
:Server 1
:wonderland.lit
:teaparty@wonderland.lit
request to chat @ teaparty
request 'teaparty' policy
retrieve applicable policy
Match FOAF and vCard ontologies
reconcile Server_1 with wonderland.lit policy
send reconciled policy with bridge axioms
check request against reconciled policy
check request against reconciled policy
propagate presence from Client 1 to teaparty chatroom

Figure 5: XMPP ontology matching.

The alignment between FOAF and vCard was used to dynamically produce OWL bridge axioms, which allow for reconciliation between policies using related, but differently named concepts. Thus, Server 2 can determine whether a client's foaf:nick has the required prefix, and subsequently is of the necessary gender, based on the "nickname" filed of his vCard. Although the scenario used a rather trivial example of matching, our design and implementation can support more complex alignments, as long as the matcher can first automatically align the ontologies.

VI. Conclusions

In this project, we have demonstrated that:

1. Policies authored in a restricted natural language format (SBVR Structured English) can be automatically converted to an executable formalism (BaseVISor rule language and OWL 2 RL) effectively.

2. Policies written in the ontology-based rule language provide an effective and flexible way to specify expressive policies that can be automatically enforced using ontology-based reasoning. The core ontologies used as the basis for domain-specific knowledge are grounded by our investigation of established security models.

3. Policies written in the ontology-based rule language can be effectively reconciled to allow for dynamic, policy-based information exchange between and an open set of XMPP servers.

4. While policy reconciliation typically requires the sharing of a common vocabulary, we have shown that effective ontology matching can be implemented to allow policy reconciliation across different (but similar) vocabularies.

Acknowledgment

This work was performed under US Army contract W91260-09-C-0037 "Security Policy Reconciliation".

References

[1] Semantics of Business Vocabulary and Business Rules v1.0. http://www.omg.org/spec/SBVR/1.0/ . May 2011.

[2] MDA Home Page. www.omg.org/mda/ . May 2011.

[3] MOF Home Page. www.omg.org/mof/ . May 2011.

[4] R. G. Ross. Basic RuleSpeak® Guidelines: Do's and Don'ts in Expressing Natural-Language Business Rules in English. http://www.rulespeak.com/en/Basic%20RuleSpeak%20Dos%20and%20Donts%20v2-2-5.pdf

[5] Business Rule Solutions, LLC. Home Page. http://www.brsolutions.com/ . May 2011M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[6] P. McDaniel and A. Prakash. Methods and Limitations of Security Policy Reconciliation. ACM Transactions on Information and System Security (TISSEC), Association for Computing Machinery, 9(3):259-291, 2006

[7] SBVR Visual Editor Home Page. http://sourceforge.net/projects/sbvrve/ . May 2011

[8] Sepiax-Web Page. http://www.sepiax.org/index.php?id=93 . May 2011

[9] M. De Tommasi, A. Corallo. SBEAVER: A Tool for Modeling Business Vocabularies and Business Rules. In Proc. 10th Int. Conf. on Knowledge-Based Intelligent Information and Engineering Systems (KES'06), LNCS Vol. 4253, 2006, 1083–1091

[10] MDT/SBVR Proposal Page. http://wiki.eclipse.org/MDT-SBVR-Proposal . May 2011.

[11] Eclipse Model Development Tools (MDT) Home Page. http://www.eclipse.org/modeling/mdt/ . May 2011.

[12] P. McNamara, Deontic Logic. *The Stanford Encyclopedia of Philosophy (Fall 2010 Edition)*, Edward N. Zalta (ed.). http://plato.stanford.edu/archives/fall2010/entries/logic-deontic/

[13] UCore Specification Page. https://www.ucore.gov/ . May 2011.

[14] B. Smith, L. Vizenor, J. Schoening. Universal Core Semantic Layer. Ontologies in the Intelligence Community Conference, 2007.

[15] Lee, R. Using New Standards to Develop IC Ontologies. In Proc. of the Fifth International Conference on Semantic Technologies for Intelligence, Defense, and Security. (STIDS'10). Fairfax, VA, USA, October 27-28, 2010.

[16] XMPP Standards Foundation. www.xmpp.org/ . May 2011.

[17] Openfire Home Page. http://www.igniterealtime.org/projects/openfire/

[18] Ignite Realtime Home Page. http://www.igniterealtime.org/ . May 2011.

[19] Spark Home Page. http://www.igniterealtime.org/projects/spark/index.jsp . May 2011.

[20] CWID 2010 UK Cross Domain Chat. Enclosure 1 to Cross Domain Chat Point Brief. July 2010.

[21] Boldon, James – Military Messaging and Secure Information Exchange Software Page. http://www.army-technology.com/contractors/navigation/boldonjames/ . May 2011.

[22] Isode Whitepaper: Using Security Labels to Control Message Flow in XMPP Services. http://www.isode.com/whitepapers/controlling-message-flow.html . May 2011.

[23] Common Information Sharing Standard for Information Security Marking: XML Implementation Implementation Guide. Office of the Director of National Intelligence Chief Information Officer. Release 2.0.3, February 2006.

[24] J. Euzenat, F. Scharffe, and A. Zimmermann, "Expressive alignment language and implementation," deliverable, Knowledge Web NoE, 2007. Available at http:// ftp//ftp.inrialpes.fr/pub/exmo/reports/kweb-2210.pdf..

[25] Kim, J. Luo, and M. Kang, "Security Ontology for Annotating Resources," Proceedings of 4th International Conference on Ontologies, Databases, and Applications of Semantics (ODBASE'05), Agia Napa, Cyprus, 2005.

A Framework for Ontology-Supported Intelligent Geospatial Feature Discovery Services

Liping Di, Peng Yue, Peisheng Zhao, Wenli Yang, Weiguo Han
Center for Spatial Information Science and Systems
George Mason University
Fairfax, Virginia 22030
(ldi, pyue, pzhao, wyang1, whan)@gmu.edu

***Abstract*—Geospatial feature discovery from remote sensing imageries is widely used in national defense and security communities. Existing methods in the geospatial image mining and feature extraction focus on the manual or automated processing of images to detect individual elementary features, such as building and highway. Such elementary features don't tell much semantic information about the features. Compound geospatial features such as Weapons of Mass Destruction (WMD) proliferation facilities are spatially composed of elementary features (e.g., buildings for hosting fuel concentration machines, cooling ponds, and transportation railways). The identity and much semantic information of a compound geospatial feature can be derived from the spatial relationship among the elementary elements. In this paper, we propose a flexible service framework for discovering compound geospatial features using an ontology-supported approach. The ontology for facilities helps find compound features that contain the specified spatial relationships among constituent features. The framework uses Web services for elementary feature extraction or access of existing elementary features, identifies facilities based on semantic descriptions of elementary feature constituents and their spatial relationships, and composes workflow-based service chains for automatic feature discovery.**

Keywords-image mining, semantic web, ontology, geospatial services, workflow, feature discovery

I. INTRODUCTION

Rapid increasing in the remote sensing capability in recent years, especially in the high spatial resolution imaging, has allowed identification of geospatial features and their changes over time. Consequently, analysis of geospatial imagery has become a promising approach for characterizing Weapons of Mass Destruction (WMD) (including nuclear) proliferation. However, the overwhelming volume of routine image acquisition has greatly outpaced the increase in the capacity of manual interpretation by intelligent analysts and has prompted automated approaches for image processing for information or knowledge generation that can be used in the decision making. An automated system, which can automatically identify geospatial features such as suspicious WMD proliferation sites in images for intelligence analysts to further investigate, can significantly reduce the workload of human intelligence analysts and increase the possibility of prompt detection of WMD proliferation.

Current methods in the geospatial image mining and feature extraction focus on the manual or automated processing of the incoming images to detect elementary visual features, such as building, highway [1-3]. Classification is often performed on per-pixel basis, although region-based characterization has received increasing attention in the recent years [4, 5]. On the other hand, geospatial images contain complex (compound) features and patterns. These features and patterns contain spatial relationships (metric, topological, etc). Traditional image analysis approaches mainly exploit image features, such as, color and texture, and, to some extent, size and shape. These image features ignore important spatial (topological) relationships [6], without which we cannot accurately discover complex (compound) features that relate to facilities used for manufactory, storage, and transportation of WMD.

In this paper, we present the concept and implementation architecture of an ontology-supported approach for discovering complex features in a service oriented environment. Web service technologies can be used to extract, access, and discover features in a distributed environment. Ontologies for complex features include semantic descriptions of elementary features and their spatial relationships. Using ontological reasoning and planning, we can decompose a detection task into a series of spatial/temporal relationship queries. The automatic execution of the series of queries to detect complex features such as possible WMD proliferation site can be performed through the workflow execution under a Service-Oriented Architecture (SOA)-based system. In the rest of the paper, we use the semiconductor manufacturing facility as an example of complex features.

The remainder of the paper is organized as follows. Section II describes the discovery of elementary geospatial features. These features can be either extracted from imagery using feature extraction services or accessed from existing feature repositories. In Section III, we discuss the ontologies for complex features, whose constituent features and spatial relations among them support the decomposition of detection tasks into workflows in Section IV. Section V describes the implementation architecture in SOA. Discussion and conclusion are given in Section VI.

II. DISCOVERY OF ELEMENTARY GEOSPATIAL FEATURES

A manufacturing facility consists of a group of elementary ground features (e.g., buildings for hosting fuel concentration machines, cooling ponds, transportation railways, fence, etc). Discovery of elementary features can be conducted by either 1) performing new feature extraction from high resolution remote sensing images or 2) accessing existing feature repositories. In both cases, the Catalogue Service for the Web (CSW) from the Open Geospatial Consortium (OGC) can be used by front-end users to find the features of interest.

There are already a significant number of algorithms for extracting elementary features (e.g., building, railways, highway, and ponds) from high spatial resolution images and other sources [1-3]. It is promising to adopt new technologies and flexible systems to make these existing algorithms capable of plug-in-and-play for on-demand feature extraction from high-resolution images.

Web services provide a promising prospect to have feature extraction done automatically over the Web. A geospatial Web service is a modular Web application that operates on geospatial data, information, or knowledge. Geospatial Web services can perform any function from simple geospatial data request to complex geospatial analysis [7]. Individual geospatial Web services can be chained together as workflows to accomplish complex tasks. Figure 1 shows an example of service components connected in the pipeline as a workflow for providing new geospatial features.

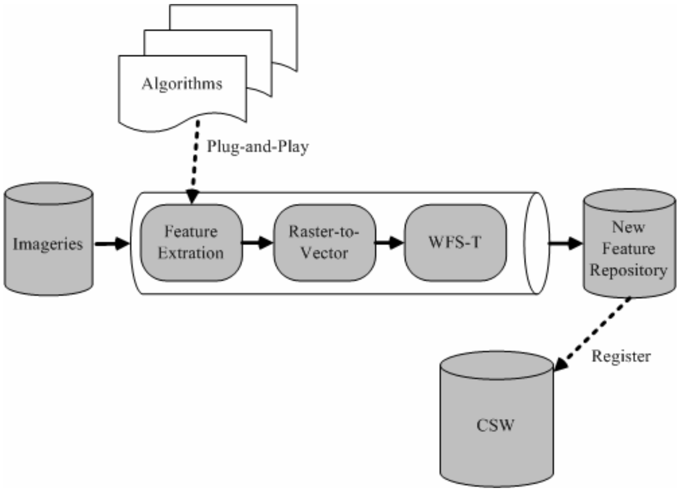


Figure 1. Feature Extraction and Registration

Most of remote sensors collect data as images in raster form. The feature extraction algorithms can recognize objects on images. Each algorithm can be provided as a feature extraction service and plugged into the workflow. The geospatial data at the feature level is in a vector form. The extraction results need to be converted into vectors using a raster-to-vector conversion service. The vector is inserted in a transactional Web Feature Service (WFS-T) and stored in a new feature repository. WFS-T is an OGC standard specification. The new features can be registered in the CSW. The feature extraction services can also follow the OGC Web Processing Service (WPS) specification. These OGC service specifications define the standard interfaces and protocols for geospatial services to ensure the interoperability of these services.

III. ONTOLOGIES FOR COMPOUND FEATURES

The development of ontologies for compound features is to encode the knowledge of the subject matter experts in a form that can be used for automated inference and retrieval of the knowledge items. In turn, automated inference and knowledge item retrieval will be used to facilitate matching between the subject area concepts and classes of the features extracted from the imagery.

Two types of ontologies are used in describing the semantics of compound features:

1) Spatial and spatiotemporal ontologies for compound geospatial features associated with manufacturing facilities. For example, spatial predicate such as surrounded by, near, and cross can be created in the ontologies.

2) Ontologies for describing semantic concepts of and assigning semantic labels to compound geospatial features and their constituents for manufacturing domain scenarios based on current and future functional and operational requirements. The elementary features, such as building, fences, bridges, railway, railway stations, and airports, can be defined here.

By combining the two ontologies together, the application ontology, such as the manufacturing facility ontology, can describe the concepts of manufacturing facility, related compound ground features and their elementary features, and spatial/temporal relationships among the member features and with the surrounding environment (surrounding ground features).

To achieve maximum flexibility of the ontology design, a layered approach can be utilized: for most general concepts we can use existing upper-level ontologies such as DOLCE or BFO [8, 9]. The next level, describing concepts specific to geographic domain and spatial relations, can be based on the existing ontologies, the SWEET ontology [10] and Ordnance Survey Ontologies [11]. SWEET Ontology can be used in parts relating to geosciences and remote sensing, and Ordnance Survey Ontologies can be used in parts relating to buildings and facilities. Ontologies pertaining to manufacturing facility can be developed by extending existing ontologies.

IV. TASK DECOMPOSITION FOR COMPLEX FEATURE DISCOVERY

When elementary features are available from either image extractions or existing feature repositories, the task of detecting a manufacturing facility in an intelligent system can be decomposed into a series of queries of geospatial features based on the spatial relationship among the features. Examples of the queries are:

- Find groups of buildings surrounded by fences
- Find a school near metro station in a specific city
- Find roads cross parks

The above queries are constructed on spatial relationships between two groups of geographic features (binary query). A more complex query may involve multiple geographic features. For example: *Find a group of buildings surrounded by fence and near a railway station*.

The binary query can be generalized as:

Find [<quantifier1>] <feature1> <operator> [<quantifier2>] <feature2> [in <spatial area>] [at <time>]

where, feature1 and feature2 are the geographic features, such as building, schools, railways, highways, airports, bridges, fences, factories, etc, which can be topologically expressed as point, line, or polygon. Only features available in the CSW are allowed here. The quantifier1 and quantifier2 put the quantification on the features, e.g., single, a group of. The operator defines the relationship between the two geospatial features. Example of operators includes "surrounded by", "near", "containing", "cross", etc.

Complex queries can be accommodated by a chain of the binary queries with the output of the previous query is the input of the next query. For example, the complex query above can be viewed as *Find (Find groups of building surrounded by fence) near a railway station.*

The decomposition process, which decomposes a detection task into a series of spatial relationship queries, can be automated through ontological reasoning and planning. A simple scenario of manufacturing facility detection is as following:

1) Users submit a request to detect an instance of facility whose concept has been described in an ontology;

2) The request is converted to a complex spatial relationship query based on the constituent features and their spatial relationship defined in the ontology for manufacturing facilities;

3) The complex query is transformed into a chain of the binary queries;

4) The chain of the binary queries is used to construct a workflow;

5) The workflow is transformed into a service chain, consisting of services and data in the CSW.

6) The workflow engine executes the service chain to generate the answer, which is displayed in the graphical user interface overlaid with the source images to show locations of matched geographic features.

Thus, the execution of series of queries to detect the possible compound features can be performed through the workflow execution under a service system. Spatial operators in the binary queries are implemented as geospatial web services. This dynamic and automatic reasoning process takes into consideration the feature semantics and service semantics when generating workflows. The workflows can be combined with feature extraction workflows for on-demand detection of facilities from high-resolution images.

V. IMPLEMENTATION ARCHITECTURE

The ontology-supported intelligent geospatial feature discovery system is a SOA-based system. The system is designed to be application neutral (i.e., it can be used to solve the similar issues in different applications, for example, either WMD proliferation sites or other facilities detection, by using different domain ontologies). Figure 2 shows the architecture of the system.

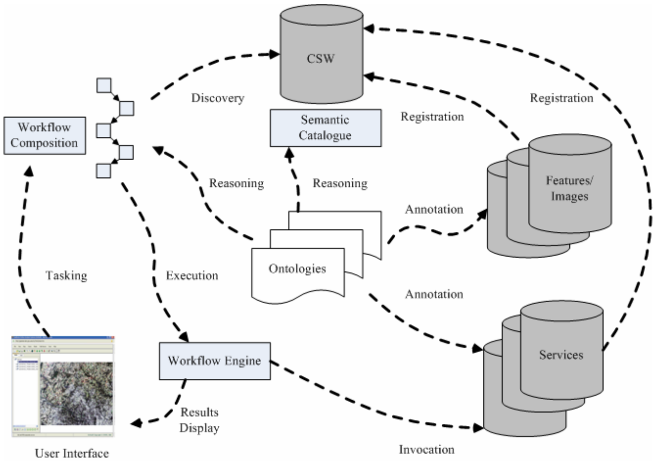


Figure 2. Implementation Architecture

A. Ontoloiges

The system can be used in different applications as long as ontologies for such applications are available. The ontologies are described using Web Ontology Language (OWL). The inference engine is used to find all relevant entailments.

B. CSW

CSW is an OGC standard specification for metadata catalogue services. Metadata for data and services are registered in the CSW for search. The semantic annotations are registered in an OGC compliant semantic catalogue to allow the semantics-enhanced discovery of available data and services [12].

C. Workflow

The facility detection task can be decomposed into a series of binary queries (spatial operations) on elementary features using the ontological reasoning and planning in the workflow composition service. The series of operations are encoded as a service chain, which will be executed by a workflow engine.

D. Data and Services

The data includes the features that have been already extracted from images. When only raw images are available, feature extraction services or workflows will be used. Feature extraction, spatial relations, and other utility services such as coordinate transformation and file format conversion can be provided. Some of them are already available in the GeoBrain project [13]. The semantics for data and services are annotated using entities from ontologies.

E. User Interface

The user interface includes both the graphical user interface (GUI) and API interface. The GeoBrain project has developed a powerful portal called GeOnAS for user interaction [14]. It can be modified in the prototype implementation here.

VI. DISCUSSION AND CONCLUSION

Currently extraction of semantic information and semantic labeling of the features in high-resolution remotely sensed imagery is a very actively developing area of remote sensing [15-17, 6]. Typically such algorithms use training data in the form of image segments with known objects and then use various statistics to match training data with the imagery. Even though effective for many purposes, such one-step approaches are likely to fail when there are subtle differences between the complex features on an image. For example, presence of an industrial chimney is a salient feature distinguishing nuclear power plant from a coal-firing plant. However, chimney is a relatively small feature in the planar view that is unlikely to produce distinguishable effect in matching statistics. The use of formally encoded semantic information in the form of ontologies can solve these problems by explicitly identifying salient features of the compound objects and their constituents.

The approach described in this paper is a two-step approach, with step 1 to identify the location and type of elementary ground features (such as building, road, etc) from high-resolution images, which has mature technology already, and step 2 to extract high-level semantic information (such as nuclear fuel concentration facilities, nuclear test sites, missile test sites, etc) through discovering compound ground features based on spatial (topological) relationships among the elementary features. The concepts and spatial relationships are described in ontologies. Therefore, ontologies are essential for system to work. However, the system is ontology independent. It can work in different domain with different domain ontologies.

The development of the system follows the SOA paradigm and using Web services and Semantic Web technologies. It allows discovery and composition of features and services in a distributed environment. The automatic discovery of compound features using workflow composition and execution can reduce the workload of human intelligent analysts and provide valuable information in decision making. The next step will be the proof-of-concept implementation, and evaluation the ontology approach.

ACKNOWLEDGMENT

This research is supported by a grant from U.S. Department of Energy (grant #DE-NA0001123, PI: Prof. Liping Di)

REFERENCES

[1] J. B. Mena and J. A. Malpica, “An automatic method for road extraction in rural and semi-urban areas starting from high resolution satellite imagery,” Pattern Recognition Letters, vol 26, issue 9, pp. 1201-1220, July 2005.

[2] J. Ahmad and CK Murali, “Automatic Feature Extraction: A Solution for Extracting the Features from High Resolution Satellite Images,” in Map Asia 2006. http://www.gisdevelopment.net/technology/rs/ma06_147abs.htm.

[3] G. Sohn and I. Dowmana, “Data fusion of high-resolution satellite imagery and LiDAR data for automatic building extraction,” ISPRS Journal of Photogrammetry and Remote Sensing, vol 62, issue 1, pp. 43-63, May 2007.

[4] A. Frome, Y. Singer, F. Sha, and J. Malik, “Learning globally-consistent local distance functions for shape-based image retrieval and classification,” in IEEE 11th International Conference on Computer Vision (ICCV 2007), pp. 1-8.

[5] X. Wang, B. Waske, and J. A. Benediktsson, “Ensemble methods for spectral-spatial classification of urban hyperspectral data,” in Geoscience and Remote Sensing Symposium (IGARSS), 2009 IEEE International, 2009, pp. 944-947.

[6] R. R. Vatsavai, B. Bhaduri, A. Cheriyadat, L. Arrowood, E. Bright, S. Gleason, C. Diegert, A. Katsaggelos, T. Pappas, R. Porter, J. Bollinger, B. Chen, and R. Hohimer, “Geospatial image mining for nuclear proliferation detection: Challenges and new opportunities,” in Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International, 2010, pp. 48−51.

[7] L. Di, P. Zhao, W. Yang, G. Yu, and P. Yue, “Intelligent geospatial web services,” in Geoscience and Remote Sensing Symposium (IGARSS), 2005 IEEE International, 2005, pp. 1229 - 1232.

[8] DOLCE, “DOLCE: A Descriptive Ontology for Linguistic and Cognitive Engineering,” Laboratory for Applied Ontology - DOLCE. Retrieved November 6, 2010, from http://www.loa-cnr.it/DOLCE.html

[9] B. Smith, “Basic Formal Ontology,” Retrieved November 5, 2010, from http://www.ifomis.org/bfo

[10] R. Raskin, “SWEET Ontology. Semantic Web for Earth and Environmental Terminology (SWEET),” Retrieved November 6, 2010, from http://sweet.jpl.nasa.gov/

[11] Ordnance Survey, “Ordnance Survey Ontologies,” Retrieved November 6, 2009, from http://www.ordnancesurvey.co.uk/oswebsite/ontology/

[12] P. Yue, J. Gong, L. Di, L. He, and Y. Wei, “Integrating semantic web technologies and geospatial catalog services for geospatial information discovery and processing in cyberinfrastructure,” GeoInformatica, vol 15, issue 2, 2011, pp. 273–303.

[13] L. Di, “GeoBrain-a web services based geospatial knowledge building system,” in Proceedings of NASA Earth Science Technology Conference 2004. June 22-24, 2004. Palo Alto, CA, USA. (8 pages, CD-ROM).

[14] W. Han, L. Di, P. Zhao, Y. Wei, an dX. Li, “Design and implementation of GeoBrain online analysis system (GeOnAS),” In Proceedings of 8th International Symposium on Web and Wireless Geographical Information System, Michela Bertolotto, Cyril Ray, Xiang Li (Eds):Lecture Notes in Computer Science (LNCS), 5373, 2008, pp. 27-36.

[15] K. W. Tobin, B. L. Bhaduri, E. A. Bright, A. Cheriyadat, T. P. Karnowski, P. J. Palathingal, T. E. Potok, et al., “Automated feature generation in large-scale geospatial libraries for content-based indexing,” Photogrammetric engineering and remote sensing, vol 72, issue 5, May 2006, pp. 531-540.

[16] S. Gleason, R. Ferrell, A. Cheriyadat, R. R. Vatsavai, and S. De, “Semantic information extraction from multispectral geospatial imagery via a flexible framework,” in Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International, 2010, pp. 166-169.

[17] R. R. Vatsavai, A. Cheriyadat, and S. Gleason, “Unsupervised semantic labeling framework for identification of complex facilities in high-resolution remote sensing images,” in 2010 IEEE International Conference on Data Mining Workshops (ICDMW), Sydney, Australia, pp. 273 – 280.

CHAMPION: Intelligent Hierarchical Reasoning Agents for Enhanced Decision Support

Ryan E. Hohimer, Frank L. Greitzer, Christine F. Noonan, Jana D. Strasburg
Pacific Northwest National Laboratory
Richland, WA

Abstract — We describe the design and development of an advanced reasoning framework employing semantic technologies, organized within a hierarchy of computational reasoning agents that interpret domain specific information. The CHAMPION reasoning framework is designed based on an inspirational metaphor of the pattern recognition functions performed by the human neocortex. The framework represents a new computational modeling approach that derives invariant knowledge representations through memory-prediction belief propagation processes that are driven by formal ontological language specification and semantic technologies. The CHAMPION framework shows promise for enhancing complex decision making in diverse problem domains including cyber security, nonproliferation and energy consumption analysis.

Keywords — Semantic Graphs, Description Logic Reasoning, Belief Propagation, Memory-Prediction Framework, Case-Based Reasoning, Ontological Engineering

I. Introduction

A major challenge for information analysis is to develop *joint cognitive systems*, described by Woods [1, 2] as systems in which humans interact with another, artificial, cognitive system. Cognitive systems are goal-directed, using knowledge about "self" and the environment to monitor, plan, and modify actions in pursuit of goals. They are both data-driven and concept-driven. Woods observed that "developments in computational technologies (i.e., heuristic programming techniques) have greatly increased the potential for automating decisions" and for "… the support of human cognitive activities…." [1] A single, integrated system was envisioned at that time that could be composed of both human and artificial cognitive systems working collaboratively to perform complex decision making tasks. In the quarter-century that has passed since this vision was described, many different types of intelligent systems and processing frameworks have been proposed and developed, though it is not clear that the vision of joint cognitive systems has been realized. The current research and development effort represents a serious attempt to bring us closer to this vision utilizing semantic modeling.

II. Background

Understanding how the human brain works is one of science's grand challenges [3]. A great deal of effort has been devoted to the development of data-driven approaches to information analysis, inspired by neuroscience, in particular the neuron. A neuron is a cell in the brain whose principal function is collection, processing and distribution of signals. These signals are propagated through networks of neurons controlling brain activity and formulating the basis for human learning and intelligence including perception, cognition and action. Artificial intelligence (AI) as a field of inquiry has been around for decades and currently encompasses a large number of subfields intersecting biology, engineering and complex systems [4-6].

Properties of biological memory systems motivate the sub-field of artificial neural networks (ANN), one type of computational model representing a bottom up or data-driven approach [7]. Feed-forward or recurrent ANNs learn by example and are able to model nonlinear systems. They require data for training the network, which is not always available. From the decision support perspective they have the disadvantage of being "opaque" to the user [8]—that is, the distribution and weights of the neural network connections are not sufficiently specified to offer insight into their operation; and this clearly doesn't facilitate collaboration of joint cognitive systems.

Machine learning is a mature field focused on programming computers to optimize performance based on past experience. The goal with this type of research is to develop general purpose systems that can adapt to new circumstances and domain knowledge [9]. A disadvantage of machine learning approaches, when coupled with human decision makers in a joint cognitive systems context, is similar to that described above for ANNs and connectionist solutions to the extent that the workings of the machine learning component are not readily understood or communicated to the human decision maker.

In contrast to these data-driven approaches, research in knowledge-based/expert systems has focused more on concept-driven or top-down reasoning. Top-down reasoning tries to mimic the brain's functions such as memory. This area of AI is concerned with thinking; how knowledge is represented symbolically and manipulated and how it contributes to intelligence.

Bayesian Network (BN) modeling approaches have become a rapidly growing area of research aimed at modeling human cognitive and decision making behavior, reflecting a perspective that use of probabilistic models and associated computational power of the Bayesian mathematical framework greatly facilitates the representation of human performance within a rational decision making framework. BN models can be viewed graphically to represent probabilistic relationships in a given domain; hence they are more readily comprehended by users. Nevertheless, there are un-answered questions regarding the appropriateness of using the Bayesian probability construct, which reflects the assumption that human decision processes may be explained in terms of rational/normative models [10].

Logic-based/rule-based systems comprise a structured collection of rules. A long-standing top-down approach is the use of logic, as represented in rule based expert systems. A major difficulty in implementing such knowledge-based systems is the difficulty of collecting expert knowledge that must be represented in the collection of rules that comprise the knowledgebase. The use of semantic web technology provides an expressive knowledge representation using ontologies, along with the application of Description Logics, which provides a formal knowledge representation language that facilitates generation of conclusions or predictions.

Unlike most problem solving techniques in artificial intelligence, case based reasoning (CBR) is memory based. Solving a problem using the classic CBR cycle involves four major components - retrieve, reuse, revise and retain (see Figure 1) [11, 12]. CBR systems are concept-driven and rely on the recognition of previously-learned (hard-coded) or experienced representations to determine the system's response to new information. A challenge for the CBR approach is the development of efficient and effective methods to search the repository of cases (stored in case memory).

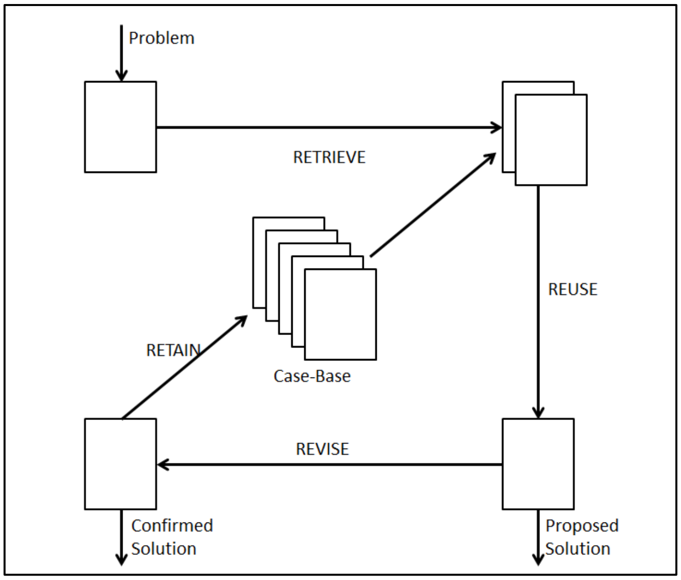


Figure 1. The CBR Cycle, adapted from [13].

A relatively recent top-down approach showing great promise is the memory prediction framework (MPF) [14, 15]. The MPF defines how the neocortex uses a feedback loop to store memory patterns which can lead to prediction of future events. These higher level concepts of cognitive processing have been applied in our work in development of the CHAMPION system.

We advance many of the aforementioned artificial intelligence concepts through extensive use of semantic technologies. With our modeling architecture, we separate domain knowledge from the reasoning framework. This is done to maintain flexibility with domain knowledge, allowing it to be updated as needed, and to ensure domain agnosticism, allowing the system to be implemented in many fields of inquiry.

III. System Design

The neocortex was the *inspirational metaphor* for the design of our reasoning framework, called CHAMPION (for Columnar Hierarchical Auto-associative Memory Processing In Ontological Networks). This metaphor serves as a representation for a functional (not structural) design adopting the following requirements :

- Stores sequences in an invariant form
- Stores sequences of patterns
- Stores sequences in a hierarchy
- Retrieves sequences auto-associatively

The CHAMPION architecture incorporates a significant variation on knowledge intense case based reasoning (KI-CBR) depicted in Figure 2. Modifications to the traditional CBR cycle were invented in order to meet the functional requirements of this metaphor.

- Instead of iteration through the case library to find a useful solution, our system uses semantic expressions to represent the criteria for a case belonging in the case library. We consider this an invariant form of a concept belonging to the set of cases.
- The functional requirement to store sequences of patterns is met by representing the problem and solution spaces in the form of semantic graphs. The nodes and edges constitute the patterns.
- The architecture uses the query/construct capabilities of SPARQL and programming pattern paradigm of "Publish and Subscribe" to implement an auto-associative mechanism.
- The domain ontology of the system addresses the functional requirement to store the concepts in a hierarchy.

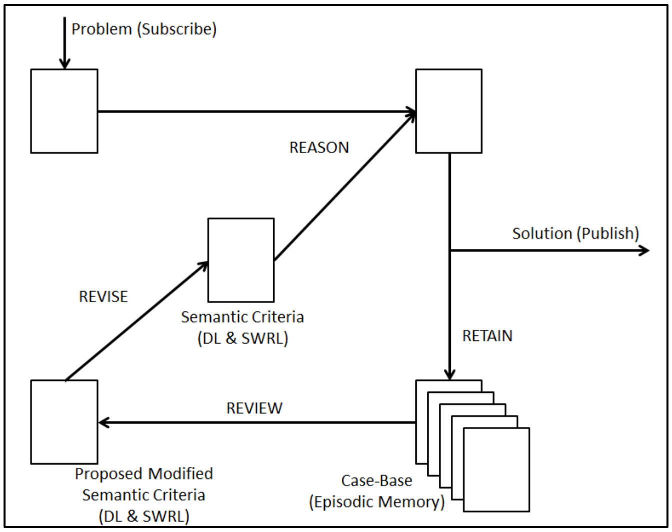


Figure 2. The CHAMPION modified CBR cycle

The CHAMPION reasoning framework consists of a hierarchy of reasoning agents called Auto-associative Memory Columns (AMCs). The hierarchy is formed as each agent subscribes to subgraphs of interest from a base graph and publishes subgraphs back to the base graph (i.e. making the base graph an inference graph).

Agents interpret data in a similar fashion as subject matter experts. The lowest level agents in the hierarchy interpret the rawest form of data, and pass their interpretation of that data up the hierarchy. Primitive data goes in the bottom and higher level interpretations come out the top.

A basic premise adhered to is the separation of the domain knowledge from the reasoning framework. If domain knowledge is hardcoded within the reasoning framework, then the framework's source code must be changed and recompiled frequently as domain knowledge is updated. Equally important is the fact that this separation of domain knowledge from the reasoning framework maintains the domain agnostic quality of the system, which enables its application to diverse problems without modification to the reasoning framework. We use the Ontology Web Language (OWL) as our knowledge representation language, to implement the ontologies and knowledgebases of the system.

The main components of the CHAMPION system, shown in Figure 3, are:

- *Ontologies*, used for representing the specialized domain knowledge.

- *Reifiers*, used for ingesting the primitive data as individuals of the types specified in the domain ontologies.

- *Memory*, used to store the facts asserted from the primitive data and the facts inferred by the reasoning system.

- *Auto-associative Memory Columns (AMCs)*, reasoning components used to interpret the data assertions and infer new assertions.

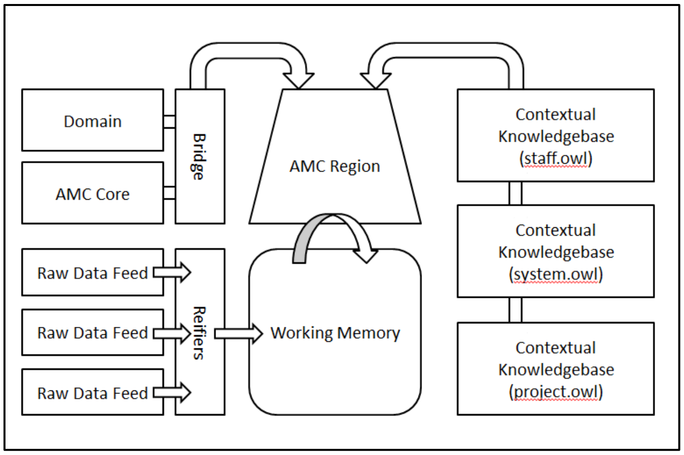


Figure 3. The components of the CHAMPION system

A. *CHAMPION Ontologies*

There are four key ontologies in the CHAMPION system, each having a unique purpose: Domain Ontology, Core Ontology, Bridge Ontology, and a collection of Rules Ontologies.

1) *Domain Ontology*

The content in the domain ontology is the knowledge of the subject matter expert in the domain of discourse to be reasoned about. It is expected that the specialized terminology of interest be captured in this T-Box ontology. If the domain of interest is Insider Threat, concepts used by experts in this field are defined here. Concepts specifically about aspects of trusted persons, their access, privileges, roles, responsibilities, and authorities would be defined. Additionally, concepts of the enterprise within which they function would be defined, such as concepts related to the infrastructure and business systems.

2) *Core Ontology*

The content of the core ontology is the knowledge of the reasoning framework and its elements. The definitions that describe what the necessary components of the AMCs are encoded into this ontology. The primary concept defined in this ontology is the AMC. The AMC is the primary reasoning agent of the framework and the class definition of the AMC is found in the core ontology.

3) *Bridge Ontology*

The bridge ontology associates concepts in the domain ontology with concepts from the core ontology. In other words, this is the place where domain concepts are assigned an AMC to reason about them.

Continuing with the Insider Threat domain, let's assume the concepts of *access* and *unauthorized access* are defined in the domain ontology as Access and UnauthorizedAccess respectively. In this example, Access is the superclass of UnauthorizedAccess. In the bridge ontology we encode that an AMC is assigned to reason about UnauthorizedAccess (the AMC class is subclassed to be an UnauthorizedAccess). The UnauthorizedAccess AMC is further defined to subscribe to

Access individuals, and publish UnauthorizedAccess individuals. Later in this paper, we will see that this is a subsumptive AMC.

4) Rules Ontologies

An AMC in the reasoning framework is to publish the appropriate assertions that are entailed in the local AMC's graph. Two governing ontologies are applied to the local AMC, 1) the domain ontology, and 2) an AMC specific ontology which contains knowledge that is relevant to the local AMC only. The consequence of having an ontology at the AMC granularity is that a rules ontology must exist for each AMC.

B. Knowledgebases

In addition to the ontologies, the following knowledgebases are required: Working Memory, AMC Knowledgebases (Binning Queue, Case Library), and a Contextual Knowledgebase.

1) Working Memory

The Working Memory knowledgebase is the semantic graph containing the state of the base-graph and the inference-graph assertions. This is the location of all the individuals from reifiers and from AMCs.

2) AMC

Each AMC has to have a local knowledgebase over which it can reason. The local knowledgebase directly imports the bridge ontology, which in turn indirectly imports the core and domain ontologies. Additionally, each AMC has a dedicated ontology that contains semantic expressions specific to this AMC. These expressions include SWRL rules that the local AMC's description logic reasoner evaluates.

3) Contextual Knowledge

Additional knowledge beyond the streaming problem data under analysis or search is stored in contextual knowledgebases. This type of knowledge needs to be accessed by the AMC in order to do informed searches or analysis. For example, to correctly reason about an activity associated with a username, the AMC must be able to access information about that username, such as the roles and access controls that are associated with that user.

C. Auto-associative Memory Columns

The analysis of real world data presents a challenge to computationally analyze very large graphs. The difficulty is not so much a *data reduction* problem as it is a *data interpretation* problem. A traditional approach to analyzing large graphs is to build the graph and then conduct reasoning over the entire graph. In contrast, the CHAMPION hierarchy of reasoners comprises a "stack" of individual AMCs which reason over the data as it is introduced into the system in much smaller graphs than the entire dataset. The larger graph structure is built as data are analyzed; this produces a dynamic belief propagation network that takes in primitive data and pushes the interpretation of that data up the hierarchy. We can think of this as interpreting the current structure in the data and simplifying with abstracting semantics. Just as we can stack the AMCs, we can stack collections (*regions*) of AMCs that address reasoning or pattern recognition for different domains. Similarly, even higher level collections of AMCs enable reasoning across such regions, providing a natural mechanism for high level information fusion and analysis.

Using a hierarchical framework of reasoners allows us to constrain the requirements of each reasoner to a narrowly-defined purpose. There is almost a one to one relationship between AMCs and the classes defined in the domain ontology. With a well-formed domain ontology, we can overcome computational intractability by performing reasoning on *subsets* of the semantic graph. Rather than implementing a monolithic reasoner that is required to reason over all the concepts represented in the semantic graph, each reasoner in the hierarchy is only required to reason about a small set of relevant concepts.

The belief propagation network performs a transformation of the low level literal inputs into higher level abstractions. Ingesting and properly formatting the input data for a given domain is performed by a *reifier*, which instantiates the input from a data source and packages the information into an OWL representation called an *individual*. In turn these individuals are instantiated in Java objects called *abstractions*. The *abstractions* are added to the *Working Memory* of the CHAMPION system.

D. Reifiers

Reifiers are responsible for asserting *individuals* (primitives) into the Working Memory via *abstractions*. Although AMCs are domain agnostic, this is not possible with the reifiers. The reifier takes in raw literal data and forms an *individual* that is defined by the domain ontology. When raw data needs to be reified, specific code is required to convert the raw data into a data-type defined in the domain ontology.

E. Provenance Information

Provenance has been defined as the description of the origins of data and the process by which it came to exist [16, 17]. Clearly this is an important requirement for the system that will facilitate the decision maker's understanding of the reasoning process. The system has two locations where provenance information can be stored. The first is in the asserted individuals added to the graph. Reified individuals (i.e. individuals from a reifier) and inferred individuals (i.e. individuals from an AMC) can have data properties asserted specifying their time and source of instantiation. The second location for storing provenance information is the episodic memory of the AMCs. Each AMC has an instantiation history of all the individuals that it has classified as being a member of its governing class. This constitutes its case library, comprising each inference graph the AMC has asserted into the base graph.

To date we have not focused on collection of provenance information. However, in future research we wish to use provenance information for two significant purposes: 1) intelligent rollback to a point of logical consistency, and 2) adaptive machine learning of higher level class resolutions based on case library analysis.

IV. An Agent's Purpose

A. *Initial Base Graph Assertions are "Primitives"*

The first assertions into the base graph are defined as "primitives." These are not primitives in the same sense as how programming languages define them, but in the sense that they are defined by a subject matter expert. These primitives are nodes that are believed to be assertions with very low uncertainty. For example, the data reified into the base graph could be computer workstation events such as security events, application events, and system events. No assumptions are made about the events; they occurred and the information is reified into the base graph. However, as reasoning agents infer new assertions based upon these primitive assertions, uncertainty can be introduced into the graph.

B. *Inference Graph Assertions are "Abstractions"*

The AMCs are in fact "classifiers". Each AMC in the hierarchy is configured by an ontology that defines classes that are the types of things in the domain of interest. In other words, the ontology contains the class definitions of the domain concepts. Class definitions are the abstract data types of the domain. Concepts are recognized by CHAMPION reasoners that have been configured to detect them. This means that for each AMC in the hierarchy there is a class definition in the governing ontology.

The purpose of each AMC is to recognize the existence of an individual of the type that belongs to its assigned class. If the individual does exist, the agent publishes the appropriate assertions.

V. The Taxonomy of CHAMPION AMCs

There are several types of AMCs in the CHAMPION system. Each AMC has the job of classifying the individuals that exist in the system. To deal with different kinds of concepts, it is necessary to define different kinds of reasoners within the AMCs. We have defined the following types of reasoning agents:

- Subsumptive
- Composite
  - Aggregate
  - Existential

We will discuss each of these in the following sections.

A. *Subsumptive Reasoning Agents*

Subsumption is rather straight forward. The knowledge representation language (OWL) used to implement our governing domain ontology specifically defines the predicates for subclassing and superclassing. A subsumptive agent examines the state of subscribed subgraphs and determines if the subgraph is subsumed by a higher level class defined in the ontology. Consider the following example:

A subsumptive reasoning agent would be used to recognize that an asserted Vehicle was in addition to being a Vehicle a Motorcycle as well. The reasoning agent would subscribe to individuals of type Vehicle, examine the state of that individual, and determine if the state of the individual meets the criteria for being a motorcycle. For instance, the Vehicle may have two wheels and handlebars, thus qualifying it as a Motorcycle. The reasoning agent would then publish the added assertion that the Vehicle was also a Motorcycle.

B. *Composite Reasoning Agents*

Composite reasoning agents are less straightforward. Unlike subsumption which is supported by explicit subclassing and superclassing predicates of standards based ontology languages, the composite reasoner examines user defined predicates to determine if the classification is valid. Subsumption only requires that a new typing assertion on an existing individual be made, not the creation of a new individual. A composite reasoner on the other hand may need to create a new named individual, not just new assertions on existing individuals.

C. *Aggregation Composite Reasoning Agents*

These agents must recognize when the requisite parts to an individual are present, and if so, create the new individual. An example of this kind of reasoning follows:

Continuing with the Vehicle example, a composite reasoning agent would subscribe to subgraphs that represented parts of a Motorcycle. These would be individuals of type Wheel and Handlebar. When the reasoning agent recognizes that all the requisite parts of a specific Motorcycle exist it creates a new individual and makes the appropriate object property assertions.

An important aspect of this aggregation process is the concept of making sure that the pieces are all parts of the same whole. In the CHAMPION system we refer to this notion as a "binning property." This property can be thought of as a Vehicle Identification Number (VIN) on an automobile. The VIN is a number that is used to keep track of the parts that belong to a specific automobile. It is not true that any four wheels, any engine, any fender, or any two bumpers sensed as inputs are the parts that make up an automobile. There has to be a mechanism to assure us that these parts all belong to the same car. This is the purpose of the binning property of a CHAMPION Composite Reasoning Agent, to make sure that the parts are recognized as being parts of a specific whole.

D. *Existential Composite Reasoning Agents*

Existential reasoning agents are very similar to aggregation reasoning agents in the fact that they have the capability to create a new individual if it is appropriate to do so. However, the aggregate reasoning agent is looking for the sum of a whole, looking to entail the existence of a thing if its necessary parts exist. An existential reasoning agent is looking to entail the existence of a thing based on evidence that it should exist. As an example of existential reasoning, if we know that a traffic ticket exists which identifies a particular license plate, we can infer that a vehicle exists. In contrast, an example of aggregation reasoning would be if we watched for vehicle parts and when we found the parts necessary to make a vehicle we could infer a vehicle exists.

The assertion that a traffic ticket exists carries little uncertainty. The inference that a vehicle exists based on the assertion of the traffic ticket carries with it a level of higher uncertainty than the existence of the traffic ticket. There could not have been a violation without the vehicle, but it may have been destroyed as a result of the violation. If we assert that it exists based on the fact that a traffic ticket refers to it, we are propagating a level of uncertainty.

VI. AMC Clockworks – Making AMCs Tick

CHAMPION AMCs comprise several components. The main component is a modified CBR mechanism. We have customized a traditional approach to CBR in order to meet the design criteria established early in our implementation.

A. *Traditional Case Based Reasoning Cycle*

A traditional CBR cycle iterates through instances of cases in a case library. As a new case is considered in traditional CBR it is compared to each of the cases in its case library. If a match is found it is considered to be a solution/match to the new case. If an exact match is not found in the case library, the closest match is modified to see if it can be made to match. If it can it is considered a solution and the modified case is added to the case library.

B. *CHAMPION's Modified Case Based Reasoning Cycle*

We chose to alter the traditional CBR cycle because the iterations through the case library to find an exact match do not fit our functional requirement to use an ***invariant form*** to characterize solutions.

The CHAMPION CBR cycle doesn't iterate through instances of cases in a case library. As a new problem case is considered it is compared to semantic expressions to see if qualifies (i.e. it belongs to the appropriate class) to be in the case library. A Description Logic (DL) reasoner is used to examine the state of the new case, if that state entails that the classification is true, the new case is added to the case library, and published to the working memory (see Figure 2).

In traditional CBR the case library is used as a repository for cases that will be iteratively compared to new input cases. This is not the purpose of the case library in our modified version of CBR. The CHAMPION system maintains the case library for the purpose of statistical analysis. The results of the statistical analysis can be used to improve the semantic expressions that define whether or not the abstractions belong in the case library.

C. *Processes of the AMCs*

The semantic expressions which define the class of objects recognized by the reasoning agents are implemented in the form of Semantic Web Rule Language (SWRL) and equivalent class expressions in OWL. The Reasoning Agents use a DL Reasoner to examine the state of the subscribed abstractions and modify the data and object properties of the abstractions.

A basic flow of the processes of an AMC:

1. Accept subscribed abstractions into local memory.
2. Acquire the requisite/relevant knowledge from contextual knowledgebases and assert into local memory.
3. Apply SWRL rules to abstractions to check and modify their state (i.e. their data and object properties).
4. Check to see if the abstraction can be classified as the targeted type of the Reasoning Agent based on equivalent class expressions in the domain ontology
5. If the DL reasoner has typed the abstraction as the targeted type, publish the abstraction to memory and add it to the case library of this agent.

The purpose of the AMCs is to process abstractions (subscribed input) and decide if it is appropriate to publish additional assertions. The additional assertions are not limited to existing individuals, meaning that the AMCs can assert new named individuals if deemed appropriate.

VII. AMC Regions

The reasoning framework arranges the AMCs in a hierarchy. The lowest levels of the hierarchy contain AMCs that subscribe to the abstractions published to the working memory by the reifiers. The AMCs of the system have a publish and subscribe relationship with working memory (see Figures 4 and 5).

When a low level AMC publishes an abstraction, a higher level AMC may be a subscriber of that type of abstraction. This is the method in which abstractions propagate up the hierarchy. As mentioned earlier, at the lowest levels in the hierarchy one expects that the abstractions contain very little uncertainty. As the AMCs are placed higher in the hierarchy the more uncertainty is likely in their output abstractions.

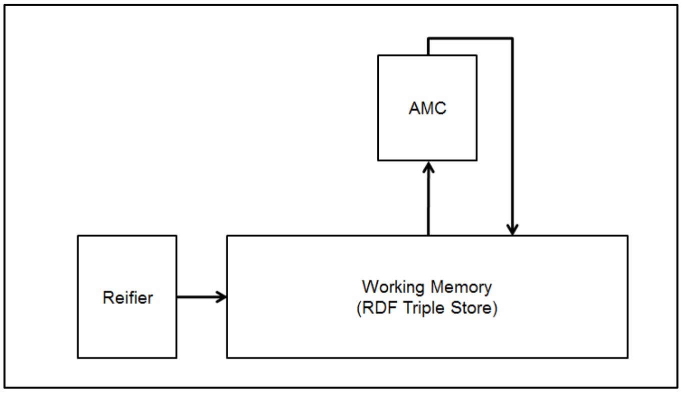


Figure 4. AMCs Publish and Subscribe to and from Memory

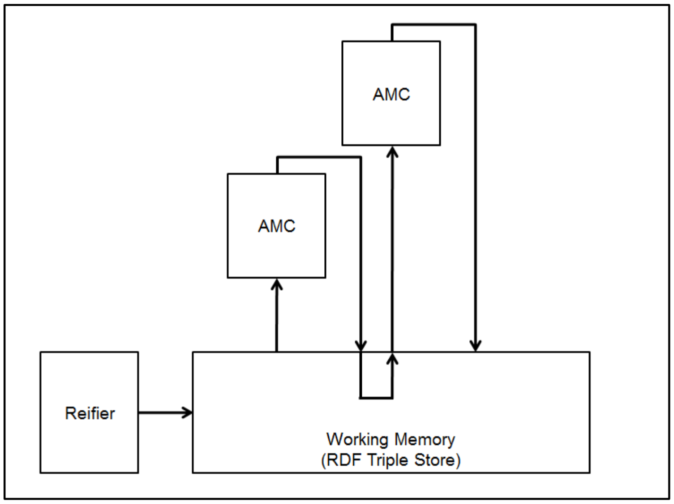


Figure 5. Abstractions passing up the AMC hierarchy

VIII. Applications

The CHAMPION reasoning framework is being applied to a variety of advanced decision making problem domains, including cyber security/counterintelligence, counterterrorism/ weapons nonproliferation, and smart grid power consumption analysis. A cybersecurity/counterintelligence application focusing on countering the insider threat is illustrative.

The insider threat refers to harmful acts that trusted individuals might carry out that may cause harm to the organization or those which benefit the individual. The insider threat is manifested when human behaviors depart from established policies, regardless of whether it results from malice or disregard for security policies. The annual e-Crime Watch Survey conducted by Carnegie-Mellon's CERT program reveals that for both the government and commercial sectors, current or former employees and contractors pose the second greatest cybersecurity threat, exceeded only by hackers; the financial impact and operating losses due to insider intrusions are increasing [18,19]. Modeling employee computer behaviors of concern using knowledge engineering methods serves as a framework to explore the insider threat. A key to the identification of an insider threat is to understand the signatures of suspicious activity and to disrupt it in its early stages. The main objective of our research is the development, validation and improvement of knowledge discovery automation tools for cyber security personnel that will significantly reduce the amount of manual analysis while simultaneously improving the quality of perceived threat indicators [20].

To create useful models, information is acquired from multiple sources including specialized reports, open literature, and subject matter experts. This information is captured via interviews with subject-matter experts (SMEs) and the development of concept maps based on domain expertise and literature analysis.

We conducted interviews of SMEs to capture information and priorities, to reveal how analysts intuitively conduct risk profiling, and to understand how they gather information about the purposes, goals and perceived risk mitigation outcomes of such activities. The information acquired is formally represented ontologically; some of the information is stored in contextual memory, and other information resides in ontologies that drive the AMCs and define the structure of the hierarchy of reasoners for this application. Figure 6 illustrates the CHAMPION system architecture within this application context.

Another interesting application for this technology is understanding nuclear proliferation. The nuclear fuel cycle is a large, complex process with many stages, dependencies, processes and signatures. In the coming year the team will use the CHAMPION framework to provide a mechanism for exploring the nuclear fuel cycle (NFC) and the logical relationships between the activities, processes, and materials involved. Working with SMEs, the team will encode the necessary knowledge into OWL to implement a proof-of-concept demonstration that will focus on a portion of the NFC. As development continues, broader coverage of the NFC will be encoded.

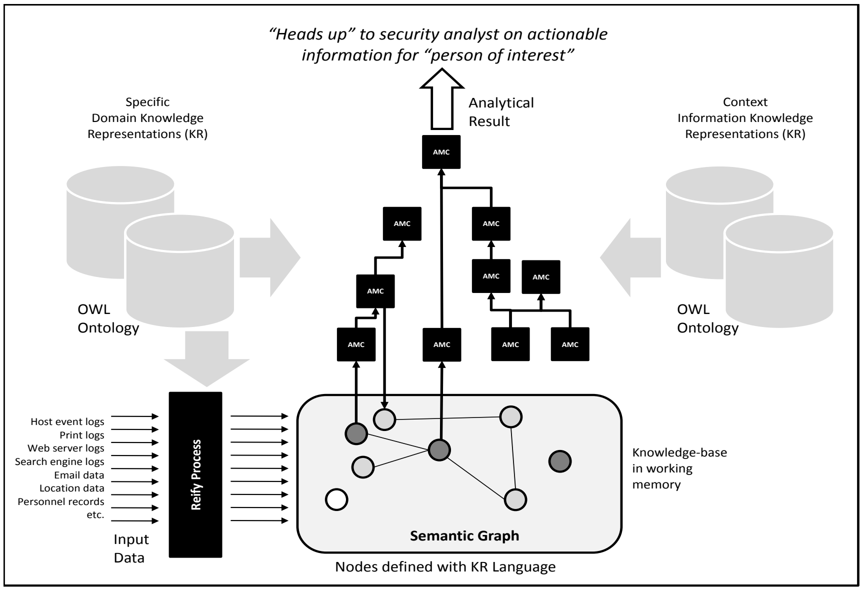

"Heads up" to security analyst on actionable information for "person of interest"
Analytical Result
Specific Domain Knowledge Representations (KR)
Context Information Knowledge Representations (KR)
AMC
OWL Ontology
OWL Ontology
Host event logs
Print logs
Web server logs
Search engine logs
Email data
Location data
Personnel records
etc.
Input Data
Reify Process
Semantic Graph
Nodes defined with KR Language
Knowledge-base in working memory

Figure 6. CHAMPION Framework in an insider threat monitoring application

IX. CONCLUSIONS

We have described a new approach to computational reasoning models that combines key aspects of belief propagation networks, semantic web, Description Logics, and Case Based Reasoning to yield a system best characterized as a memory-prediction framework. This framework is functionally modeled after an interpretation of how the neocortex performs pattern recognition. It is implemented as a hierarchy of reasoning agents that retain certain critical functional requirements that produce a domain-independent model that may be applied to a variety of decision making problems.

Earlier in this paper, we compared several extant approaches to problems in AI and noted the drawbacks of using rational decision making models to characterize human performance, such as represented in typical BN models that rely on probability theory constructs. Similar issues apply to models that apply other forms of probabilistic models such as subjective expected utility theory. Famous research programs conducted by Kahneman and Tversky [e.g., 21] demonstrate that human decision making is not rational and is rather characterized by the use of heuristics (or influenced by cognitive biases) that do not yield optimal decisions. The use of heuristics—and what has been described by Kahneman [22] as "system 1 cognitive processes" – exploiting intuition and experience rather than procedural knowledge – is sometimes cited as a critical survival mechanism that accounts for expert decision making by firefighters and other highly experienced individuals who do not have time to systematically calculate and compare outcomes of alternative responses [23]. A conceptual model that reflects this view is the "Recognition-Primed Decision Making Model" (RPDM) offered by Gary Klein and collaborators [24]. In this regard, the basic structure of the CHAMPION reasoning framework, rooted in the notion of the memory-prediction system, is very compatible with this view of expert decision making. Indeed, the CHAMPION framework represents one method of implementing an operational version of a RPDM model. It is our hope that such a model, fortified by recent computational methods adopted from semantic Web technologies, will provide a major advancement in realizing the vision for joint cognitive systems for decision support.

ACKNOWLEDGMENT

The authors wish to thank the CHAMPION team - Shawn Hampton, Samantha Curtis, Alex Gibson, Michael Henry, Dan Johnson, Gary Kiebel, Peter Neorr, Patrick Paulson, Yekaterina Pomiak, and John Schweighardt; and the Information and Infrastructure Integrity Initiative at Pacific Northwest National Laboratory which supported this work. The Pacific Northwest National Laboratory is operated by Battelle for the U.S. Department of Energy under Contract DEAC06- 76RL01830. PNNL Information Release Number PNNL-SA- 82834.

REFERENCES

[1] D. D. Woods, "Cognitive technologies: the design of joint human-machine cognitive systems," AI Magazine, vol. 6, pp. 86-92, 1985.

[2] D. D. Woods, Joint Cognitive Systems: Patterns in Cognitive Sytems Engineering. Boca Raton, FL: Taylor & Francis, 2006.

[3] Institute of Medicine Forum on Neuroscience and Nervous System Disorders. From Molecules to Minds: Challenges for the 21st Century. Washington, DC: National Academy of Sciences, 2008.

[4] B.G. Buchanan, "A (very) brief history of artificial intelligence," AI Magazine, vol. 26, pp. 53–60, 2005.

[5] N.J. Nilsson, Artificial Intellugence: A New Synthesis. San Francisco, CA: Morgan Kaufmann Publishers, Inc., 1998.

[6] R. Chrisley, ed. Artificial Intelligence: Critical Concepts, vols. 1-4. London: Routledge, Taylor & Francis Group, 2000.

[7] D.J.C. MacKay. Information Theory, Inference, and Learning Algorithms. Cambridge: Cambridge University Press, 2003.

[8] P. Smolensky, "On the treatment of connectionism," Behavioral and Brain Sciences, vol. 11, pp. 1-23, 1988.

[9] T.G. Dietterich, "Machine Learning," Annual Reviews in Computer Science, vol. 4, pp. 255-306, 1990.

[10] M. Jones and B.C. Love, "Bayesian fundamentalism or enlightenment? On the explanatory status and theoretical contributions of Bayesian models of cognition," Behavioral and Brain Sciences (in press).

[11] R. Lopez De Mantaras, et al., "Retrieval, reuse, revision and retention in case-based reasoning," The Knowledge Engineering Review, vol. 20, pp. 215-240, 2005.

[12] I. Watson and F. Marir, "Case-based reasoning: a review," Knowledge Engineering Review, vol. 9, pp. 355-381, 1994.

[13] A. Aamodt and E. Plaza, "case-based reasoning: foundational issues, methodological variations, and system approaches," AI Communications, vol. 7, pp. 39-59, 1994.

[14] A. Nouri and H. Nikmehr, "Hierarchical bayesian reservoir memory," Proceedings of the 14th International CSI Computer Conference (CSICC'09), pp. 582-587, 2009.

[15] J. Hawkins and S. Blakeslee. On Intelligence. New York: Henry Holt and Company, 2004.

[16] Buneman, P., S. Khanna, and W.C. Tan. 2001. Why and where: A characterization of data provenance. International Conference on Database Theory (ICDT), 316-330.

[17] Simmhan, Y.L., B. Plale, and D. Gannon. 2005. A survey of data provenance in e-Science. ACM SIGMOD Record, 34(3), Sept. 2005.

[18] CSO Magazine, U.S. Secret Service, Software Engineering Institute, CERT Program at Carnegie Mellon University and Deloitte. 2010 CyberSecurity watch survey - survey results.

[19] M. Keeney, et al. Insider Threat Study: Computer System Sabotage in Critical Infrastructure Sectors. U.S. Secret Service and Carnegie-Mellon University, Software Engineering Institute, CERT Coordination Center. 2005.

[20] F. L. Greitzer and R. E. Hohimer, "Modeling human behavior to anticipate insider attacks," Journal of Strategic Security, vol. 4, pp. 25-48, 2011. doi:10.5038/1944-0472.4.2.2.

[21] D. Kahneman and A. Tversky, "On the psychology of prediction," Psychological Review, vol. 80, pp. 237-251, 1973.

[22] D. Kahneman, "A perspective on judgement and choice: mapping bounded rationality," American Psychologist, vol. 58, pp. 697-720.

[23] G. Klein. Streetlights and Shadows: Searching for the Keys to Adaptive Decision Making. Cambridge, MIT Press, 2009.

[24] G.A. Klein, "A recognition primed decision (RPD) model of rapid decision making," in GA Klein, J Orasanu, R Calderwood and CE Zsambok, eds. Decision Making in Action: Models and Methods. Norwood, NJ: Ablex, pp. 138-147, 1993.

Speech acts and Tokens for Access Control and Provenance Tracking

Fabian Neuhaus
National Center for Ontological Research

Bill Andersen
Highfleet, Inc.

Abstract—**In many applications, ontology-based technologies will be only only be successful if they support access control and provenance tracking. In this paper we present a novel approach to implementation of both access control and provenance in deductive information systems. A key feature of our approach is the explicit representation of speech acts as well as sentence tokens that are used to encode propositions. These are used to define *SupportedBy*, a kind of entailment relationship between sentence tokens and propositions. User queries are phrased in terms of the SupportedBy relationship and augmented by user-dependent security and provenance constraints. We note that the introduction and treatment of SupportedBy makes the resulting logic an instance of a *labeled deductive system*, as developed by Gabbay.**

I. Introduction

To date, ontology-based technologies have been so far applied most successfully in domains like biological research where the available knowledge meets two important requirements. First, there is a network of trust among users and builders of the knowledge base that it represents a true picture of reality. Second, the knowledge in the knowledge base is open in the sense that anybody who is using the system can access all the information in the knowledge base.

However, many potential applications ontology-based systems require multi-level security access control along with 'need-to-know' restrictions. Examples include the management of the information exchanges within an engineering production network, patient data within a hospital management system, and data analysis within an intelligence agency. In addition, any application of knowledge representation technologies such applications would require reasoning with information that might turn out to be wrong, either by mistake or by ill intent. The available information might even turn out to be logically inconsistent. Within this context, it is almost as important to keep track of who provided a piece of information as keeping track of the information itself. This enables to evaluate the quality of a given information item by checking its consistency with information provided by independent sources.

Provenance tracking comes in two flavors: *hearsay tracking* and *IT processing tracking*. Hearsay tracking is concerned with the chain of 'retellings' of a piece of information before it is entered into an information technology (IT) system; e.g., in 'Novak reported that a senior official claimed that Plame suggested that Wilson travels to Niger' the basic proposition 'Wilson travels to Niger' is embedded in three layers of 'retellings'. IT processing tracking is concerned with how information is processed within a given set of IT systems. After the information is entered into an information system, it might be processed in various ways. It can be copied, recoded, or used for automatic reasoning, among others. It is important to track these processes because they can potentially cause false verifications.

In this paper we discuss the features of an ontology language that can support provenance tracking. This approach, which is based on work described in [8] and [1], has already been implemented successfully in Highfleet's XKS deductive database system. This paper clarifies the relationship of linguistic tokens and speech acts in the analysis of provenance tracking, and widens the scope of the analysis to cover hearsay tracking. As in previous work, we take a logical-ontological approach that considers (1) the entities required for an adequate accounting of access control, (2) provenance in information systems, and (3) the logical machinery that is needed to get the intended result. Hence, we will not discuss in this paper the details of the implementation.

In the next section we provide an extended example from the intelligence community that illustrates the kind of problems we intend to address in this paper. Afterward we discuss the ontological categories involved in our solution. In the last section we present a first draft of our account.

II. Example

In this paper we discuss our approach with the help of the following scenario. Assume that an Afghan source of a U.S. intelligence agency reports that Al Qaeda has obtained a nuclear weapon. The information is represented in the knowledge repository A and classified as top secret. The information about the supposed location is shared with another US agency, but the source of the information is not shared. The second agency stores the information within their knowledge repository B and classifies it as secret. Assume further, that in the same timeframe the New York Times (NYT) reports that Maulana Masood Azhar claimed in an interview that Al Qaeda has obtained a nuclear weapon. This is recorded in the knowledge repository C, which contains information collected from newspapers and other publicly-available sources. As a result, all three repositories contain (in some sense) the same information, namely that Al Qaeda has obtained a nuclear weapon. However, it is classified differently in the knowledge repositories A, B, and C (as top secret, secret, and unclassified, respectively). To further muddy the water, assume that repository B also contains reports from other sources that claim that if Al Qaeda has a nuclear weapon then it has obtained it from Pakistan; and that in addition no Pakistani nuclear weapon is missing. Thus, repository B contains conflicting information.[1]

Assume an analyst queries an information system with access to all three knowledge repositories with the following request: *Provide all independent records that support that Al Qaeda possesses weapons of mass destruction.* For the sake of simplicity, let's further assume that the knowledge repositories contain no other relevant information. This scenario provides several challenges: (i) The correct response from the system depends on the clearance of the analyst. If the analyst has no access to classified information, he should receive one answer, namely the one from the NYT report in repository C. The fact that the information provided by the Afghan source is classified should not prevent the analyst from accessing the information based on news reports, although in some sense it is the *same* information. (ii) If the analyst has access to top secret information, the system should provide two answers and not three: the NYT report and the original record in repository A. The system should not provide the record in knowledge base B, since it is based on the one in repository A, and thus does not provide independent verification. (iii) Inferring that the report by the Afghan source and the NYT article is about weapons of mass destruction requires logical reasoning with content embedded in some additional information about provenance. (iv) The available information is logically inconsistent; a fact that seems to render the classical entailment relationships useless.

We have addressed the first two challenges in [1]. In this paper we extend our solution to hearsay provenance tracking and address the problem of inconsistent information in more detail.

III. Ontology of access control and provenance

Our approach to a theory of access control is ontological rather than procedural.[2] By first examining and fixing the relevant kinds of entities involved in access control and provenance, we hope to provide a firm foundation for an evolving formal theory for handling these phenomena in information systems.

A. Information, Proposition, Sentence Types, and Tokens

According to the U.S. government the object of access control is information [9]. While an analysis of the ontological nature of information is beyond the scope of this paper, we are convinced that according to any reasonable account of information (e.g., [12]) a unit of information is an 'abstract entity' in the same sense that integers or geometric shapes are abstract entities. That is, they do not have a spatio-temporal location and are not participating in causal interactions that shape our physical world. Based on this view, it is hard to imagine how we might control directly access to anything abstract. What we can control, though, is access to physical entities that *encode* information. For example, it is impossible to lock a piece of information in a safe for the same reasons it is impossible to lock up the integer 3. We are able, however, to lock up a paper document that encodes the information. In the case of IT systems the computational mechanisms have causal influence over objects that are encoded ultimately as patterns of electrons or some other physical mechanism. We argue that the objects of access control are thus spatio-temporal objects that participate in the causal structure of information systems.

Although we are confident that our approach can be extended to information encoded in images, video, audio and other like forms of common digitally encoded media, but in this paper we will focus on information systems dealing with propositions encoded in formal language expressions. A *proposition* is a unit of information that is either true or false. A well-formed expression of a (formal) language that expresses a proposition is a *sentence type*. We note that 'formal language' is not intended to be restricted to logical languages but is intended to cover any kind of syntax including graphical notations, tables, tree structures, and barcodes.

Since sentence types and propositions are abstract entities they cannot be objects of access control and provenance tracking for the reasons given above. In contrast, *sentence tokens* are physical entities that instantiate sentence types [13]. The same sentence type can be instantiated by a large range of physical objects; a sentence token on a printed newspaper is a distribution of ink, in the case of a spoken sentence the token is a complex movement of air, and in the case of information systems the tokens are arrangements of electric charges in a chip.

Different tokens of the same types might not only differ with respect to the kind of material they consist of and other physical qualities, but, more importantly for our purposes, different tokens of the same type can differ with respect to their security properties: one encoding of a proposition P might be unclassified while another encoding of the same proposition one is classfied.

A sentence token might come into existence by accident. If you spill your coffee on a sheet of paper and it reads "It was the best of times", then this distribution of coffee on this sheet of paper is an instance of the sentence type. However, usually sentence tokens are brought into existence by a person in an attempt to communicate with somebody else, a *speech act* [2], [11]. A speech act is a kind of intentional act performed by a person (the speaker) typically involving one or more listeners, a sentence token, a proposition, and the *illocutionary force* of the speech act. For example, utterances of the assertion 'You are late', the question 'Are you late?' and the command 'Be late!' involve the same proposition but vary in their illocutionary force – the first makes a statement about reality, the second seeks verification, and the third seeks to bring about the truth of a proposition. While these examples might suggest that illocutionary force is aligned with syntactic

[1] For the sake of simplicity, we do not treat time explicitly and just assume that the statements are valid during the same time period.

[2] The Bell-La Padula security model is one example where secure states of a system are defined by a state machine model [3].

distinctions in English, this is not the case. E.g., the utterance of 'I'll be back' might be a promise, a prediction, a warning, or a threat – depending on the circumstances and the intentions of the speaker. This example shows that while under normal circumstances the proposition of a speech act is straightforwardly encoded in the sentence token, the illocutionary force might be harder to determine. For the sake of simplicity, we will in the following widely ignore the differences between the illocutionary forces of the various types of speech acts and treat them either as assertive speech acts or as queries. For example, promises, warnings, and threats will all be treated equally as assertions.

We are concerned with speech acts for two reasons. First, if a sentence token in an IT system is the result of an entry by a human, then the sentence token is the result of a speech act. Second, we need to deal with sentence tokens that encode propositions about speech acts. In our example an analyst has read in the NYT that Maulana Masood Azhar claimed that Al Qaeda owns a nuclear weapon, and creates a corresponding entry in a knowledge repository. Creating this entry is a speech act by the analyst. The token in the knowledge repository does encode the proposition that the NYT performed a reporting speech act. The propositional content of the speech act by the NYT is that Maulana Masood Azhar has performed another speech act, namely an announcement. The proposition of that last speech act is that Al Qaeda owns a nuclear weapon. Thus, the propositions of all three speech acts are nested within each other.

B. Manipulation of tokens in IT systems

By IT system we mean a physical object that is capable of (1) accepting information encoded in tokens of some appropriate language and (2) accepting and responding to queries posed as tokens in some appropriate language with the result being a *release* of tokens encoding the query response. In this paper we are interested in IT systems with access control, that is systems that allow access to stored information only through specified processes and through no other means.

On a token-based view of access control, the policies that guide whether a given token can be released by an IT system is based on its access control properties. Thus, we must account for the causal history of tokens in an information system from the moment that information bearing tokens enter a system to when (other) tokens are released from the system. This causal history will take the form of a chain of events (copying, synthesis, and recoding) that make new tokens from old ones. Depending on the type of event, properties relevant to access control will need preservation.[3]

In this paper we consider a security labeling system that consists of a totally-ordered set of *levels* L and a set of partially ordered *compartments* C. Each token is assigned a security level and a (potentially empty) set of compartments. Security levels express the sensitivity of a given piece of information. Compartments are used to limit access channels independent of the security levels. The partial order on the set of compartments ranks the compartments along their specificity (e.g., the compartment *Al Qaeda* would be more specific than the compartment *terrorist group*). Ontologically speaking, security levels and compartments are social artifacts that are dependent upon a community of agents that mutually agrees to the storage and access of tokens using the labeling system.

IV. Formalization of access control and provenance

A. The representation in a formal language

In this section we will sketch an axiomatic approach that allows us to reason under multi-level security access control and enables provenance tracking. We are not suggesting that the logical language below is supposed to be used within a knowledge repository, nor do we suggest that the end users of the system shall be confronted with such a language. Our main concern is that the features of the implemented knowledge representation language enable queries and logical reasoning in a way that supports access control and provenance tracking as described here.

As mentioned in the introduction, Highfleet has already implemented a system with these features successfully. However, the goal of this paper is not describe a specific implementation or to discuss how the approach we are suggesting can be implemented efficiently. Our goal is just to outline the underlying logical-ontological approach. For this reason we just assume that we have a reasoner that supports a very expressive language, at least as expressive as IKRIS Knowledge Language (IKL) [5], [6] extended by two modal operators: $\Diamond$ is read as 'it is logically possibly true that' and $\Box$ is read as 'it is logically necessarily true that'.[4] IKL is an extension of CLIF which itself is the interchange format of the ISO standard Common Logic [7]. CLIF differs from many first-order languages by not assigning a fixed arity to its predicates and by adding sequence variables to the language. In the following we will use x, y, z as ordinary first-order variables and s, s_1, s_2 as sequence variables – variables that range over finite sequences of objects.

One basic idea of our approach is to treat tokens that reside in the repositories and the speech acts they encode as first class citizens in the domain of discourse. In this way the so-called 'metainformation' about security and provenance can be treated in the same logical framework as regular object-level information. Let's return to the example from the introduction. The knowledge repository A of the first agency contains a token that expresses *Source007 asserted on October 20th, 2011 that Al Qaeda owns nuclear weapons*. The most important aspects of assertive speech acts are its speaker as well as the proposition that is asserted. The example

[3] We discuss IT systems and their boundaries, as well as the the manipulation of tokens within IT systems in greater length in [1].

[4] The intended semantics is the following: $\Box A$ is true if and only if A is logical truth of classical first-order logic. $\Diamond A$ is true if and only $\sim A$ is is not a logical truth of classical first-order; i.e. A is satisfiable. The details of the extension of IKL by these modal operators are beyond the scope of this paper.

includes also the date of the speech act. Potentially other relevant information about the speech act might be available (e.g., its location or the listeners that participated in it). This ‘metadata’ about the speech act needs to be distinguished from the ‘metadata’ about the token that encodes the speech act itself. Examples for ‘metadata’ about tokens include the type of the token (e.g., record, audio file, picture), the name of the repository where the token resides, the security classification of the token, its security compartments, and the person who created the entry in the system, the date when the entry was created, a list of people who accessed the information in the system, and so on.

This ‘metadata’ can be represented in the same framework as the ‘normal’ data in the following way, where ‘token001’ is a name of the record that resides in the repository A:

Record(token001) &
ResidesIn(token001) = repository_A &
ClassifiedAs (token001 top_secret) &
Compartment (token001 alQaeda_cmpt) &
Compartment (token001 proliferation_cmpt) &
Tok1 *CreatedBy(token001) = agent1234* &
PropositionalContent(token001) =
  (that ($\exists x$ *(AssertionAct(*x*)* &
  Speaker(x *source007)* &
  Date(x*) = 20.10.2011)* &
  PropositionalContent(x*) = (that(*
    $\exists y$ *(Owns (alQaeda* y*)* & *NuclWeap(*y*)))))*

We use IKL’s mechanism for expression of propositions – the that-operator. It is applied to a formula and the result is a name that refers to a proposition. Its logical counterpart in IKL is a syntactic variant of a truth-predicate: if p is a proposition, then the formula (p) is the assertion of the proposition. Thus, $(A \leftrightarrow ((\text{that } A)))$ is a logical truth in IKL, for any formula A.

In our example, the agency that owns repository A shares the information with another agency. As a result a ‘write down’ token is created within repository B; that is the propositional content of the speech act encoded in token001 is preserved, but the additional information about the speech act is removed. As a result the newly created token is reclassified as secret. The information about the token in repository B can be represented in the following way:

Record(token002) &
ResidesIn(token002) = repository_B &
ClassifiedAs(token002 secret) &
BasedOn(token002, token001) &
Tok2 *ResidesIn(token001) = repository_A* &
PropositionalContent(token002) =
  (that ($\exists x$ *(AssertionAct(*x*)* &
  PropositionalContent(x*) = (that(*
    $\exists y$ *(Owns (alQaeda* y*)* & *NuclWeap(*y*)))))*

The fact that the entry in knowledge base B originated from repository A is expressed explicitly by asserting that token002 is based on token001 and that token001 resides in the repository A. Using the ’BasedOn’ relationship in this way enables provenance tracking across the knowledge repositories, and enables a reasoner to detect that token001 and token002 do not provide independent verification of the information concerning Al Qaeda’s access to nuclear weapons. This example points to a further advantage of our approach – namely that it provides a principled way of performing ’write-down’ operations, enabling more flexible sharing of information without compromising sensitive meta-information. Such operations are typically not allowed by traditional security models, e.g., [3].

The entry based on the NYT report is distinguished from the previous examples by an additional layer of indirectness: the token encodes a proposition about a speech act about a speech act. Each of the speech acts has a propositional content and a speaker; in this example the dates of the speech acts are provided as well.

Record(token003) &
ResidesIn(token003) = repository_C &
ClassifiedAs(token003 unclassified) &
PropositionalContent(token003) =
  $\exists x$ *(AssertionAct(*x*)* &
  Speaker(x nyt) &
Tok3 *Date(*x*) = 23.10.2011* &
  PropositionalContent(x*) =*
    (that ($\exists y$ *AssertionAct(*y*)* &
    Speaker(y MasoodAzhar) &
    Date(y*) = 22.10.2011* &
    PropositionalContent(y*) = (that(*
      $\exists z$ *(Owns (alQaeda* z*)* & *NuclWeap(*z*)))))*

B. The support relationship

We now add another relationship ”SupportedBy” between a proposition and a sequence of zero or more records. The goal of this relation is to capture not only the propositional content that is captured in one record, but what is logically entailed by the sequence of these records. One problem we need to address is that in the framework of a classical logic a contradiction logically entails any proposition. Assume we have an ontology-based information system with a classical reasoner. In our example, the knowledge base B contains records that encode the following propositions: (i) Al Qaeda owns a nuclear weapon; (ii) if Al Qaeda owns a nuclear weapon, then Pakistan is missing it, and (iii) the Pakistanis are not missing a nuclear weapon. If we were to provide these propositions in an unaugmented way to this system, the reasoner would ‘use’ these contradictory assumptions to prove any query – and thus the IT system would become useless. Our goal is to enable limited reasoning with contradictory information, but to prevent the system from ‘exploding’.[5] This

[5] This goal is the driving force behind the development of paraconsistent logics. Since we are defining a (object language) relationship between tokens and propositions our goal is slightly different than the one in paraconsistent logic which is concerned with the (meta language) entailment relationship. The ”SupportedBy” could be briefly characterized as a paraconsistent variant of the strict implication with a closure on embedded propositions.

is achieved with the help of the two modal operators $\Diamond$ and $\Box$ introduced above.

Instead of SupportedBy((that A), s) we write A[s] as a shorthand. In particular, we write A[ ] to express that A is supported by the empty sequence. We axiomatize the SupportedBy relationship recursively with the following axiom schemata:

Ax1 $(\mathit{Record}(x)\ \&\ \Diamond(\mathit{PropositionalContent}(x))) \rightarrow$
$\mathit{SupportedBy}(\mathit{PropositionalContent}(x), x)$

Ax2 $A \rightarrow A[\,]$

Ax3 $(A[s_1]\&B[s_2]\&\Diamond(A\&B)) \rightarrow (A\&B)[s_1\ s_2]$

Ax4 $(A[s]\&\Box(A \rightarrow B)) \rightarrow B[s]$

Ax5 $(\Diamond A\ \&\ (\exists x((\mathit{AssertionAct}(x)\&$
$(\mathit{PropositionalContent}(x) = (\mathit{that}\ A))))[s]) \rightarrow A[s]$

Ax1 expresses that every record supports its (own) propositional content – under the condition that assertion of the propositional content is possibly true. Further, every proposition that is already known to be true is supported by the empty sequence (Ax2). According to Ax3 the following holds: if a proposition A is supported by a sequence of records s_1 and a proposition B is supported by a sequence of records s_2 and (A & B) is possibly true, then the proposition (A & B) is supported by the sequence that is the result of concatenating s_1 and s_2. Note that if A and B are logically contradictory, it is not possible that (A & B) is true; thus in this case A[s_1] & B[s_2] do not imply (A & B)[s_1 s_2]. Without this constraint a sequence of assertions of contradictory information would support every proposition because, as discussed above, in classical logic a logically false formula will entail any formula. Ax4 expresses the following: if the sequence s supports a proposition A and A necessarily implies B, then the sequence s also supports the proposition B. The axiom ensures that a sequence of records does not only support a conjunction of their propositional contents but also the logical consequences of the propositions. Ax5 ensures that a token does not only support the proposition it encodes but also all propositions that are embedded in that proposition – provided that they are logically possible.

We made a few simplifications in these axioms. First of all, we axiomatized SupportedBy based on sequences of tokens. Sequences that consist of the same components in different order are different sequences; e.g. (token001 token005) and (token005 token001) are two different sequences. Consequently, an IKL reasoner will consider them as different answers to a query. However, for SupportedBy the order of the sequence elements does not matter, any permutation is as good as another. Further, the approach delivers sequences that contain tokens that are not necessary to support the proposition. For example, the answer (token001 token005) would be a valid answer to query Que1, in spite of the fact that token001 supports the proposition on its own and token005 does not contribute anything to the answer. Thus, the axioms as presented above would deliver redundant answers. It is possible to avoid these problems, but for the sake of brevity we present a simplified account.

C. Reasoning with SupportedBy

The support relationship is used to enable queries for information that support a given hypothesis. In the rest of this section we will show how that works with the help of the example from the introduction. However, within this section we will ignore that Tok2 is based on Tok1; this will be the subject of the next section. Let's assume that the system has access to an ontology that either contains or logically entails the background information Bgnd1: A nuclear weapon is a weapon of mass destruction (WMD).

Bgnd1 $\forall x(\mathit{NuclWeap}(x) \rightarrow \mathit{WMD}(x))$

In our example, the analyst is interested in the question whether Al Qaeda possesses WMD. For starters, we can represent the query 'Find all sequences of records that are supporting the proposition Al Qaeda owns WMD.' in the following way:

Que1 $\exists x(\mathit{Owns}(\mathit{alQaeda}\ x)\ \&\ \mathit{WMD}(x))[?s]$

Note that IKL itself does not provide any convention to express queries; hence, we use question marks in front of variables to mark variables to be bound by a reasoner.

When the analyst enters the query Que1 into the system, it tries to find a sequence of tokens that enables it to prove the query. For example, the system would try to prove that token001 supports this proposition. Example1 shows how a proof could look like.

Example1

1) $\forall x(\mathit{NuclWeap}(x) \rightarrow \mathit{WMD}(x))[\,]$
2) $\exists x(\mathit{AssertionAct}(x)\ \&\ \mathit{Speaker}(x\ \mathit{source007})\ \&$
$\mathit{Date}(x) = \mathit{20.10.2011})\ \&\ \mathit{PropositionalContent}(x) =$
$(\mathit{that}(\exists y(\mathit{Owns}(\mathit{alQaeda}\ y)\&\mathit{NuclWeap}(y)))[\mathit{token001}]$
3) $\Box((\exists x(A\ \&\ B\ \&\ C\ \&\ D)) \rightarrow \exists x(A\ \&\ D))$
4) $\exists x(\mathit{AssertionAct}(x)\ \&\ \mathit{PropositionalContent}(x) =$
$(\mathit{that}(\exists y(\mathit{Owns}(\mathit{alQaeda}\ y)\&\mathit{NuclWeap}(y)))[\mathit{token001}]$
5) $\Diamond(\exists y(\mathit{Owns}(\mathit{alQaeda}\ y)\&\mathit{NuclWeap}(y)))$
6) $\exists y(\mathit{Owns}(\mathit{alQaeda}\ y)\ \&\ \mathit{NuclWeap}(y))[\mathit{token001}]$
7) $\Diamond(\forall x(\mathit{NuclWeap}(x) \rightarrow \mathit{WMD}(x))\&$
$\exists y(\mathit{Owns}(\mathit{alQaeda}\ y)\ \&\ \mathit{NuclWeap}(y)))$
8) $(\forall x(\mathit{NuclWeap}(x) \rightarrow \mathit{WMD}(x))\ \&$
$\exists y(\mathit{Owns}(\mathit{alQaeda}\ y)\ \&\ \mathit{NuclWeap}(y)))[\mathit{token001}]$
9) $\Box((\forall x(\mathit{NuclWeap}(x) \rightarrow \mathit{WMD}(x))\ \&$
$\exists y(\mathit{Owns}(\mathit{alQaeda}\ y)\ \&\ \mathit{WMD}(y)))$
$\rightarrow \exists x(\mathit{Owns}(\mathit{alQaeda}\ x)\ \&\ \mathit{NuclWeap}(x)))$
10) $\exists x(\mathit{Owns}(\mathit{alQaeda}\ x)\ \&\ \mathit{NuclWeap}(x))[\mathit{token001}]$

Line 1 of the proof is an immediate consequence of Bgnd1 and Ax2. Line 2 follows from Tok1 and Ax1. Line 3 is a modal theorem schema, and the proposition of line 2 matches the antecedent. Thus, line 3 in combination with Ax4 can be used to remove the information about the speaker and the date from line 2, the result is line 4. Lines 5, 7, and 9 are theorems under the intended interpretation of the modal operators. Lines 4, 5, and Ax5 give rise to line 6. Lines 1, 6, 7, and Ax3 entail line 8 of the proof. Line 8, 9, and Ax4 entail line 10. Q.E.D.

Example1 shows that (token001) (the sequence consisting only of token001) is one possible answer to query Que1. In a similar fashion one can prove that (token002) and (token003) answer the query. While Example1 is admittedly rather simple, it is sufficient to show how these proofs work and what role the axioms play: the 'background information' that is provided to the system as truths (e.g., nuclear weapons are WMD), lead to SupportedBy-statements with an empty sequence by axiom Ax2. Formulas that express the content of records (like Tok1) lead to SupportedBy-statements via axiom Ax1 that contain lists with only one element. In our example we use these axioms only once each, but in more complex examples one would have to use these axioms repeatedly. The resulting SupportedBy-statements can be combined with more complex ones with the help of axioms Ax3. The role of Ax4 is to ensure that consistent lists of tokens support all logical consequences of their propositions. If a proposition is embedded in another proposition, then Ax5 (in combination with Ax4) allows us to show that the former proposition is supported by the same sequence of tokens as the latter.

Example2

1) $\forall x(\mathit{Owns}(\mathit{alQaeda}\ x)\ \&\ \mathit{NuclWeap}(x)) \rightarrow$ $\mathit{Misses}(Pakistan\ x))[\mathit{token004}]$
2) $\sim\exists x(\mathit{Misses}(Pakistan\ x)\ \&\ \mathit{NuclWeap}(x))[\mathit{token005}]$
3) $\sim\exists x(\mathit{Owns}(\mathit{alQaeda}\ x)\ \&\ \mathit{NuclWeap}(x))[\mathit{token004}\ \mathit{token005}]$

In Example1 the proposition in the last line is only supported by one token, but a proposition can be supported by an arbitrary long list of tokens. Let token004 encode the proposition 'if Al Qaeda owns a nuclear weapon, then Pakistan misses it', and let token005 encode 'Pakistan is not missing a nuclear weapon'. By applying Ax1 we get the first two lines of Example2. They in combination with Ax3 and Ax4 entail the third line: an example for a proposition supported by two records.

Note that it is not possible to combine the last lines of Example1 and Example2 with Ax3, because $\Diamond(A\ \&\ {\sim}A)$ is not provable, for any given formula A and any given set of consistent assumptions. This is an example how the axioms of the SupportedBy relationship block unwanted reasoning with inconsistent information.

So far, in all examples that we discussed the provenance of the information has been ignored. So what is the benefit to represent the speech acts explicitly within the formulas? First of all, the analyst does need to know who provided the information and whether it is hearsay or the result of direct observation. In addition, it allows us to support queries that mix 'normal' queries with 'metainformation' about security classification and provenance. For example, the analyst might ask the following additional query: *Find all top secret records that involve an assertion by Masood Azhar that entail the existence of nuclear weapons.* This query can be represented as follows:

Que2 $\mathit{Record}(?x)\ \&$
$\mathit{ClassifiedAs}(?x\ \mathit{top\_secret})\ \&$
$(\exists y(\mathit{AssertionAct}(y)\ \&\mathit{Speaker}(y\ \mathit{MasoodAzhar})\ \&$
$\Box((\mathit{PropositionalContent}(y)) \rightarrow$
$\exists z\mathit{NuclWeap}(z)))[?x]$

D. IT processing tracking and access control

In the last section we addressed hearsay tracking which is one aspect of provenance tracking. We did not address provenance tracking of tokens within IT systems. In our example token002 resides in knowledge repository B. It is based on token001, which resides in knowledge repository A. If an analyst queries the system that has access to the knowledge repositories A and B, then the information of token002 has to be ignored, since it provides no independent confirmation of the information. However, it might be the case that knowledge repository B but not repository A is available for queries; for example because of technical difficulties or because the agency of the analyst is not allowed to use repository A. In this case the system is supposed to use the information encoded in token002.

To support this functionality, for example, the query Que1 would have to be rephrased in the following way: *Find sequences s of tokens that support the proposition that Al Qaeda owns WMD, which meets the following additional requirement: there are no tokens y, z such that: (a) y resides in a repository that is available, (b) z is an element of the sequence s, (c) z is a copy of y, and (d) the sequence that is the result of replacing all occurrences of z in s by occurrences of y supports the proposition.*

Access control adds another layer of complexity. The query changes from 'Find me a sequence of tokens that support X' to 'Find me a sequence of tokens *that the user has access to* that support X'. Whether a user has access to a given token is determined by its security level and its security departments. We provide a detailed analysis of how to represent process tracking and access control in [1].

As we have seen in the case of Que2, the treatment of information about provenance on the same level as any other information enables queries that otherwise would not be possible. So far we looked at use cases that provided information for end-users. However, it is also useful for system administrators. For example, Que3 represents the query: *Find all secret records within repository B that are based on top secret records of repository A.*

Que3
$\mathit{Record}(?x)\ \&$
$\mathit{ResidesIn}(?x) = \mathit{repository\_B}\ \&$
$\mathit{ClassifiedAs}(?x\ \mathit{secret})\ \&$
$\exists y(\mathit{BasedOn}(?x\ y)\ \&$
$\mathit{ResidesIn}(y) = \mathit{repository\_A}\ \&$
$\mathit{ClassifiedAs}(y\ \mathit{top\_secret}))$

$$\textbf{Tok4}\quad\begin{array}{l}
\mathit{Record(token005)}\ \&\\
\mathit{ClassifiedAs(token005\ secret)}\ \&\\
\mathit{PropositionalContent(token005)} =\\
\quad \mathit{(that\ (\exists x\ (QueryAct(x)}\ \&\\
\qquad \mathit{AskedBy(x\ analyst1234)}\ \&\\
\quad \mathit{Date(x) = 12.11.2011}\ \&\\
\quad \mathit{PropositionalContent(x) = (that(}\\
\qquad \mathit{\exists y\ (Owns\ (alQaeda\ y)}\ \&\ \mathit{NuclWeap(y)))))}
\end{array}$$

Another use case is to track which analyst accesses which data from which system. For example, assume that an analyst asks the query whether Al Qaeda owns nuclear weapons. At the same time the system answers the query, the system could generate Tok4, which records that the analyst asked a query, its date, as well as its propositional content. This information can be used to monitor who accesses which data from what sources and on what security level. It is also enables systems administrators to recognize if two independently working analysts are interested in the same content.

E. Non-propositional information

As mentioned above, the main focus of this paper is propositional information. However, it seems that our approach of treating tokens as first-class citizens in the domain of quantification could work not only for records but also for other tokens, for example pictures and audio files. Here is an example how one could represent the information about a picture that shows Maulana Masood Azhar visiting Baba Saab in Kandahar.

$$\textbf{Tok5}\quad\begin{array}{l}
\mathit{Picture(token006)}\ \&\\
\mathit{ResidesIn(token006) = repository\_A}\ \&\\
\mathit{ClassifiedAs\ (token006\ top\_secret)}\ \&\\
\mathit{CreatedBy(token006) = agent1234}\ \&\\
\mathit{Source(token006) = source007}\ \&\\
\mathit{LocationDepicted(token006\ babaSaab)}\ \&\\
\mathit{About(token006\ MasoodAzhar)}\ \&\\
\mathit{About(token006\ Shrine)}
\end{array}$$

Security classification, the location of the token, and other 'meta-information' are provided in the same way as in previous examples. The main difference is that pictures do not encode propositional content. To tag the picture with keywords we are using the 'About' relationship (and subtypes of About like LocationDepicted). Since IKL lacks a syntactical distinction between predicates and individual constants, the second argument of the About-relationship can be filled by an individual (e.g, Masood Azhar) or a type (e.g., shrine). We are planning to further investigate the potential of our approach for the representation of non-propositional information in the future.

F. Treatment as a labeled deductive system

We note briefly here that the logic we have described may be considered a type of *labeled deductive system* [4]. The concept of a labeled deductive system is a generalization of the traditional notion of a logical system in which the consequence relation is defined relative to a system (algebra) of *labels* that modulate consequences that may be drawn in such systems. These systems have the advantage of incorporating what are traditionally viewed as meta-logical concepts (e.g. the rules for creating a proof) into the object language. Not only does this make, in many cases, for a more elegant description of the logic under consideration, it provides a unified description for logics that seem dissimilar on the surface but are in fact quite similar in terms of their underlying behavior.

The logic we describe can be considered a labeled deductive system for three reasons. First, the (sequences of) sentence tokens act as the labels of the system. Second, the operations on the labels encoded in the axioms for the SupportedBy relation define an algebra over the labels. Finally, the entailment relation for the system depends both on the content as well as the labels.

In fact, the application of labeled deduction to access control was proposed by Obrst and Nichols in [10], wherein they suggest (but did not develop) a labeled deductive system that operates over security labels in defining the consequence relation. Their proposal differs from our approach in two major ways. Our system quantifies over tokens, and enables multiple tokens of the same type to co-exist. Further, we pay specific attention to speech acts that assert tokens into a information system.

V. Discussion and future work

In this paper, we presented an ontologically-motivated approach to multi-level access control and provenance for information systems. We extended our previous work by widening the scope of the analysis to different use cases, most importantly hearsay provenance. Critical to our analysis is the role of linguistic tokens as the fundamental bearers of information and as the only entities capable of playing the causal role required to enforce access controls and track provenance within IT systems. These linguistic tokens might be bearers of information about speech acts with propositional content; these are used to enable hearsay provenance. We offered a formalized example of reasoning with provenance under multi-level access control. While the presentation was limited to access control and provenance in systems using overt logical reasoning processes, we would argue that the approach is applicable generally to information systems of all kinds (e.g., relational database systems or web-services).

In the future we are planning to extend this work to a theory of access control and provenance for non-overtly linguistic information bearing objects, such as audio, images, or video, and to account for effects of intentional degradation of information for "write-down" releases of information.

Acknowledgements

We thank Leo Obrst for his helpful comments and suggestions.

References

[1] B. Andersen, F. Neuhaus. An Ontological Approach to Information Access Control and Provenance. In P. Costa, K. Laskey, L. Obrst (eds.): *Proceedings of the 2009 International Conference on Ontologies for the Intelligence Community* Fairfax, VA, USA, October 21-22, 2009. http://CEUR-WS.org/Vol-555/paper7.pdf.

[2] J.L. Austin. How To Do Things With Words 2nd Ed. Harvard University Press, Cambridge, 1975.

[3] D.E. Bell. Looking Back at the Bell-La Padula Model In Proceedings, *Annual Computer Security Applications Conference*, Tucson, 2005.

[4] D. Gabbay. Labelled Deductive Systems; Principles and Applications. Vol 1: Introduction. Oxford University Press, 1996

[5] P. Hayes. IKL Guide. http://www.ihmc.us/users/phayes/IKL/GUIDE/GUIDE.html

[6] P. Hayes, C. Menzel. IKL Specification Document. http://www.ihmc.us/users/phayes/IKL/SPEC/SPEC.html

[7] ISO/IEC 24707. Information technology – Common Logic (CL): a framework for a family of logic-based languages.

[8] F. Neuhaus, B. Andersen. The Bigger Picture – Speech Acts in Interaction with Ontology-based Information Systems. In M. Okada, B. Smith (eds): *Interdisciplinary Ontology* Vol. 2 (Proceedings of the Second Interdisciplinary Ontology Meeting), 2009, 45-56.

[9] United States Office of the Director of National Intelligence. Intelligence Community Directive Number 501. January, 2009.

[10] L. Obrst, D. Nichols. Context and Ontologies: Contextual Indexing of Ontological Expressions. Poster: AAAI 2005 Workshop on Context and Ontologies. AAAI, Pittsburgh, PA, 2005.

[11] J. Searle. Speech acts: An Essay in the Philosophy of Language. Cambridge University Press, New York, 1970.

[12] C.E. Shannon, W. Weaver. The Mathematical Theory of Communication. Urbana, Ill.: University of Illinois Press, 1975.

[13] L. Wetzel. Types and Tokens. The Stanford Encyclopedia of Philosophy (Winter 2008 Edition), Edward N. Zalta (ed.), URL = http://plato.stanford.edu/archives/win2008/entries/types-tokens/

Finding and Explaining Similarities in Linked Data

Catherine Olsson
Raytheon BBN Technologies and
Massachusetts Institute of Technology
catherio@mit.edu

Plamen Petrov, Jeff Sherman, Andrew Perez-Lopez
Raytheon BBN Technologies
Arlington, VA
{ppetrov,jsherman,aperezlo}@bbn.com

Abstract—**Today's computer users and system designers face increasingly vast amounts of data, yet lack good tools to find pertinent information within those datasets. Linked data technologies add invaluable structure to data, but challenges remain in helping users understand and exploit that structure. One important question users might ask about their data is "What entities are similar to this one, and why?" or "How similar are these two entities to one another, and why?". Our work focuses on using the semantic content of linked data not only to facilitate the process of finding similar entities, but also to produce automatically-generated and human-understandable explanations of what makes those entities similar. In this paper, we formulate a definition of an "explanation" of similarity, we describe a system that can produce such explanations efficiently, and we present a methodology to allow the user to tailor how "obvious" or "obscure" the provided explanations are.**

I. Introduction

Today's world is a world of data. As technology advances, it becomes easier and easier to collect and store vast amounts of data. Much of this data can be viewed in terms of nodes with properties and relationships, or edges, among those nodes — that is to say, it can be represented as a graph. Once a dataset has been represented in a graph format, such as with Semantic Web [1] or other linked data technologies [2][3], it can easily be combined with data from different sources. In this way, linked data allows already-vast datasets to be readily combined and connected, giving users and programs access to more data than ever before. The challenge, then, is in making sense of this data.

Some of the data analysis questions that are emerging include the following: How does one entity in the linked data relate to another entity, possibly derived from a different source? How does a given entity relate to the rest of the data? What are the similarities between two entities, and why? There are also related data search and retrieval questions to be tackled, such as "Find all entities similar to this entity" and "Find groups of entities that are similar to each other."

To solve these problems, work has been done at Raytheon BBN Technologies (BBN) to devise a similarity measure called the Structural Semantic Distance Measure (SSDM), which leverages both the structural and semantic content of linked data to find similar entities. SSDM is based on SimRank [4], a highly domain-general similarity measure with an efficient approximate calculation. SSDM improves on SimRank by incorporating the semantic content of edge labels, and by achieving greater independence of ontological choices.

Raw numerical similarity scores provide very little insight to users about what those scores mean, so users often want an *explanation* of how a score should be understood and interpreted. In this paper, we formulate a definition of an "explanation" of an SSDM score that ensures that the explanation is both human-understandable and well-grounded in how SSDM scores are calculated. We also describe a system that can produce such explanations efficiently.

Additionally, not all users will desire the same level of detail in their explanations. Therefore, we present a methodology for allowing the user to tailor how "obvious" or "obscure" the provided explanations are. We expect that users who are investigating an unfamiliar domain will prefer "obvious" explanations that refer to common and well-known properties, while expert users will prefer "obscure" explanations that shed light on less well-known relationships and details.

A. Motivation

Our work is motivated by a number of problems in the intelligence and military research communities. Many of those problems are ubiquitous and have direct translation to business intelligence, logistics, and planning. Take, for example, a model of a large organization C with its associates, their interactions, locations they visit, resources they use or produce, and events in which they participate. Given this information, one could explore the stated relationships among the constituents of C, such as "show all transactions that involve person X." Beyond these simple information-retrieval tasks, analysts might want to examine more complex (or less crisply defined) interactions. For example, "show all associates similar to Y" could be a very useful query when trying to learn more about person Y. Finally, given a subset $S = s_1, s_2, ...s_n$ of members in the organization C, which might represent a group that is suspect of participating in nefarious activities, a query like "show all subsets of C similar to S" might be an excellent way to discover other suspicious clusters in the organization.

Note that in the example above we did not have any a priori knowledge of the organization other than its structure (which in general is a directed graph) and the elements for which we were searching. In particular, we did not assume any hierarchy, types of relationships present, or any statistical properties of the graph. It was also important that the queries were phrased in a general way using the word "similar" to indicate a degree of likeness, but not (necessarily) an exact match. Such problems occur every day both in the military, intelligence, and defense communities as well as in the business and civilian worlds.

We have structured our algorithms and methodologies to be applicable to any data expressing entities and relationships between entities. Notably, much of the data encountered in the military and intelligence domains deals with entities and the relationships between them, and can therefore benefit from our contributions.

Throughout this paper, we include examples from the movie industry, drawn from a popular and widely accessible dataset about movies, actors, directors, film genres, and so on [5]. One can easily find direct analogies between this data and the types of data encountered in the intelligence and defense domains.

II. Background

The general problem of similarity is twofold: first, to construct a measure of pairwise similarity so that a meaningful similarity score can be calculated for any pair of entities in a dataset; and second, to devise a method for efficiently retrieving the entities that are most similar to a given entity.

In this section, we discuss a similarity measure developed at BBN called the Structural Semantic Distance Measure (SSDM). SSDM is an extension of existing work on calculating similarity over unlabeled, directed graphs. The contribution of SSDM is to incorporate the *semantic* content contained in edge labels, and to achieve a greater independence of ontological choices for edge labeling.

Our work on SSDM builds off the SimRank algorithm by Jeh and Widom [4]. We chose to base our work on SimRank for the following reasons:

- SimRank is domain-independent in that it can be applied to any data representing relationships between entities. This is in contrast with domain-specific similarity algorithms, such as those that can only be used to compare documents [6], ontological categories [7], or some other domain-specific data type.
- SimRank can be computed efficiently in approximation, even over very large datasets, in contrast with measures that rely on Singular Value Decomposition or other computations that scale poorly [8].
- The approximate computation of SimRank can not only determine the similarity between two entities efficiently, but can also generate a list of entities that are most similar to a given entity.
- The computation behind SimRank can be understood on a conceptual level, which makes it possible to explain the similarity score by referring directly to the computation performed. This would not be possible using a similarity measure that relied on more abstract calculations.
- SimRank looks beyond an entity's immediate neighborhood and features when determining similarity, which enables it to incorporate a broader scope of information about the structural context of entities.

All these positive attributes are retained in SSDM, along with several additional improvements.

A. SimRank

SimRank is based on the intuition that "Two entities are similar if they are related to similar entities." While this statement may seem trivial at first, it leads directly to a simple mathematical definition of similarity: the similarity score between two entities is the average pairwise similarity of their neighbors, scaled by a decay factor.

Consider the example in Fig. 1. Intuitively, one would imagine that Movie 1 and Movie 2 should be similar, because they have two actors in common and they are both in the same genre. Additionally, Director 1 and Director 2 should be similar even though they have no immediate connections in common, because they directed similar movies. SimRank captures and formalizes this intuition.

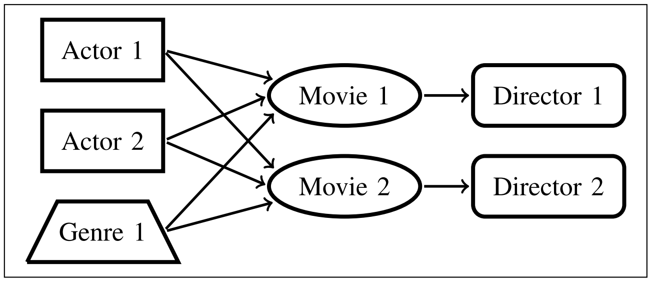


Fig. 1: This figure depicts relationships that exist between entities in a movie dataset. Director 1 and Director 2 have no immediate neighbors in common, but they are similar because they are related to similar movies

Each pair's similarity is dependent on many other pairs, which may seem to be a barrier to computing their scores. Fortunately, this barrier is readily surmountable. On small datasets the system can be solved with an iterative algorithm, and on large datasets it can be solved using an efficient approximate method outlined by Fogaras and Rácz in [9]. Our implementation of the SSDM calculation is based on this efficient approximate method.

The algorithm outlined by Fogaras and Rácz relies on the mathematical notion of a *random walk* through a graph, in which an abstract walker steps from node to node through the graph by following random edges [10]. In the original SimRank paper, Jeh and Widom observed that the SimRank score of two nodes can be approximated from the *expected meeting time* of two random walkers starting at those two nodes; a higher expected meeting time corresponds with a lower SimRank score. Fogaras and Rácz used this observation to develop an efficient and scalable algorithm for calculating similarity scores.

In the algorithm proposed in [9], one random walker is initialized per node in the graph, and each walker moves along one edge per time step. To reduce the amount of computation required, walkers are allowed to *converge* at their meeting point, and are thenceforth treated as a single walker without loss of correctness in the approximate calculation of expected meeting times. Fig. 2 demonstrates how walkers converge. Once the maximum number of steps has elapsed, the run is halted. Repeated runs are performed and the data are aggregated.

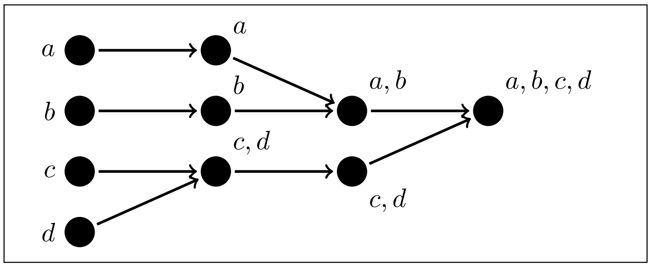


Fig. 2: Walkers a, b, c, and d begin as independent walkers. As they walk (shown progressing from left to right), they meet one another. c and d meet at the first time step and converge. a and b meet next. Finally, on the far right of the diagram, all walkers have converged.

Additionally, to repair some deficiencies with the original formulation of SimRank, walkers in Fogaras and Rácz's algorithm are incentivized to converge if they are near one another. This is accomplished by randomly permuting all vertices in the graph at the start of each time step, with each walker stepping to the neighbor with the smallest index in the permutation.

The end result of one run of Fogaras and Rácz's scalable SimRank algorithm is a *fingerprint graph* which encodes the first meeting times for each pair of walkers. Several runs are conducted, and the fingerprint graphs are compiled into a larger *fingerprint database*. The fingerprint database is precomputed and can be efficiently queried thereafter, either to retrieve a similarity score between any two nodes, or to retrieve the set of nodes with a similarity score greater than some threshold with any given node.

The implementation we devised for calculating the SSDM retains the basic structure of the computation described above, including converging random walkers and permutation-based convergence incentivization.

B. SSDM — The Structural Semantic Distance Measure

The Structural Semantic Distance Measure developed at BBN is closely related to SimRank as described above, with two key enhancements: SSDM incorporates the semantic content of edge labels, and SSDM is independent of ontological choices — namely, which edge directionality each proposition should have.

Whereas SimRank is a measure over unlabeled directed graphs, SSDM incorporates edge labels. This makes it well-suited to any data in subject-predicate-object format, such as RDF or other linked data; subjects and objects are equivalent to nodes in the graph, and predicates are equivalent to labeled edges. The most important use of edge labels is in the way expected meeting time is calculated. The original SimRank computation defines "meeting time" as the time step when two walkers step to the same node, at which point they converge. The SSDM computation has a stricter condition on convergence: namely, when two walkers meet, it only counts as a convergence if they arrive at the same node on the same step *and* they have traversed *identical sequence of predicates* to that node. The reasoning behind this modification is that the semantic meaning of edge labels is critical to the similarity calculation. For example, two entities `A` and `B` may both be related to a third entity `C`, but they are certainly not similar if the relations in question are `A isA C` and `B isNever C`.

Additionally, SimRank only allows similarity to propagate along in-edges, which means that the original computation of SimRank only allows walkers to step backwards (that is, from objects to subjects). This makes SimRank highly dependent on ontological choices, because it is an arbitrary choice in a directed graph whether each label should be phrased in the forward or reverse direction. For example, the relation `A isComponentOf B` could be equally expressed as `B hasComponent A`; the choice of which direction is used is arbitrary and can vary from dataset to dataset. Choosing and enforcing consistent edge directionality is a difficult issue in ontologies in general, so we did not want SSDM to be heavily dependent on arbitrary edge direction choices. As a result, SSDM allows walkers to walk both directions.

Note that allowing walks in both directions requires us to distinguish a walker traversing `A isComponentOf B` from a walker traversing `B isComponentOf A`, as these two steps have very different semantic meanings. Therefore, in SSDM, it is not enough for walkers to simply have traversed identical predicates in order to converge; they must have traversed those predicates *in the same direction* (in or out).

To illustrate the conditions on convergence required by SSDM, consider the example of calculating the similarity between two movies, *War of The Worlds* and *Gladiator*, as shown in Fig. 3. Suppose Walker 1, starting from *War of The Worlds*, traverses the predicates $directed_{in}$, $directed_{out}$, $hasActor_{out}$ to reach Harrison Ford. If Walker 2 follows the same predicates in the same order to reach Harrison Ford, as is shown in Path A, then the two walkers will converge with a meeting time of three steps. If Walker 2 instead follows a different sequence of predicates, such as $hasActor_{out}$, $hasActor_{in}$, $hasActor_{out}$ as shown in Path B, it will not converge with Walker 1 because the two walkers did not follow identical predicates to get there.

SSDM was designed as a domain-independent similarity measure that would easily account for the semantics of labeled graph data without being dependent on ontological choices. This ability to incorporate semantic information is especially important in domains with rich semantic context, and allows SSDM to capture semantic nuances that are missed by less sophisticated similarity measures. The significance of this extra information in the measure is as-yet unassessed, but we expect that SSDM should perform better than SimRank for semantic graphs. In short, SSDM is an efficient, semantically-grounded, and ontology-independent algorithm for discovering similar entities in a linked dataset.

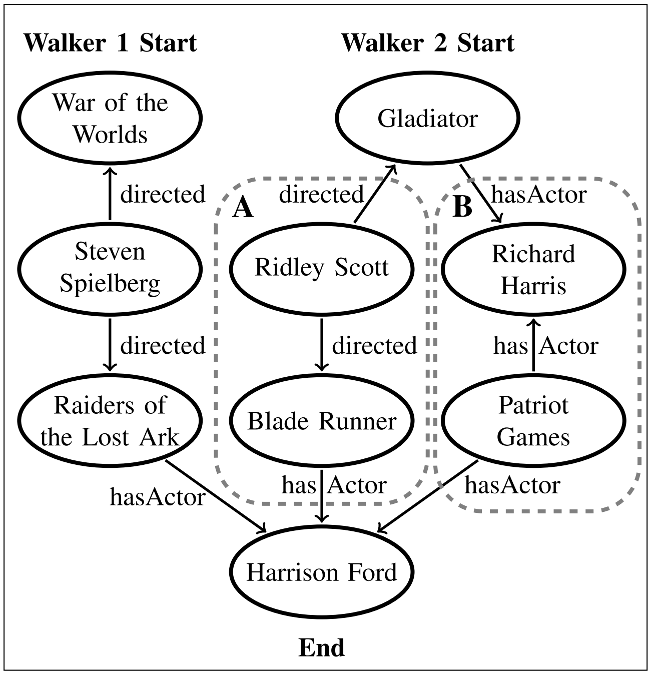


Fig. 3: This diagram depicts two possible ways — path **A** and path **B** — by which Walker 2 could meet Walker 1 at the Harrison Ford node. If Walker 2 takes Path **A**, they will converge. If Walker 2 takes Path **B**, they will not.

III. Explaining Similarities

A similarity measure is useful for ranking items or pairs of items, but a numerical score alone gives little insight into *why* two entities are similar. The user can easily retrieve the similarity score for two entities but is then left wondering: *what about* those entities and their relations caused them to receive a high or low similarity score? What is the nature of their similarity? In addition, users may want more or less depth in the explanations provided.

In order to enable users to answer this question, we sought to build a system that could provide *explanations* for similarity scores. Our three main contributions to the area of similarity explanations are as follows:

1) We formulated a definition of an "explanation" for a similarity score that is human-understandable, as well as appropriately grounded in the way that the similarity score was originally calculated.
2) We wrote a program to efficiently produce such explanations.
3) We further developed a methodology for biasing explanations towards either more "obvious" or more "obscure" facts.

IV. Definition of an Explanation

A good explanation of a similarity score must be both human-understandable and grounded in the original calculation of the similarity measure. An explanation that is not human-understandable is hardly an explanation at all, and an explanation that is generated by a computation which is entirely unlike the original calculation of the similarity score can hardly be considered an explanation of why that score was produced.

Recall that the SSDM computation calculates similarity scores based on repeated runs of converging random walkers. The faster two random walkers tend to converge, the higher the similarity score of their starting nodes will be.

It follows that for an explanation of a score to be well-grounded in how SSDM scores are calculated, it must somehow elucidate *where* and *how* walkers from the nodes in question tend to converge, and whether these convergences tend to happen rapidly or whether the walks are long. In order for this information to also be human-understandable, it must be relatively concise.

For these reasons, we decided that an explanation should consist of a brief *list of common convergence points*, along with a handful of concise *chains of statements* per convergence point describing the "best" relationships linking each starting node to that point. The "best" relationships may be the shortest chains of statements, or the ones that tend to be traversed most frequently, or (as explained later) relationships that are appropriately obvious or obscure. Figure 4 shows an example of an explanation of this form.

```
John Williams
  {A New Hope, hasMusicContributor, John Williams}
  {The Empire Strikes Back, hasMusicContributor,
    John Williams}

Harrison Ford
  {A New Hope, hasActor, Harrison Ford}
  {The Empire Strikes Back, hasActor, Harrison Ford}

Star Wars (Film Collection)
  {A New Hope, inCollection, Star Wars}
  {The Empire Strikes Back, inCollection, Star Wars}

  {Revenge of the Sith, hasSequel, A New Hope}
    {Revenge of the Sith, inCollection, Star Wars}
  {A New Hope, hasSequel, The Empire Strikes Back}
    {A New Hope, inCollection, Star Wars}
...
```

Fig. 4: An excerpt from an explanation for the similarity between *Star Wars Episode IV: A New Hope* and *Star Wars Episode V: The Empire Strikes Back*, showing both one-relationship chains and multi-relationship chains

By defining an "explanation" this way, we ensure that users are presented with a coherent explanation of *where* and *how* walkers from the nodes in question tend to converge. Additionally, such explanations are also readily understandable as explanations of what commonalities the nodes have, and how they are related to each commonality.

V. Our Approach

The most obvious way to extract an explanation for a computation seems to be to inspect the path that the computation followed to obtain its result. Unfortunately, in the case of our

SSDM calculation, such a strategy is inadequate. We have established that we would like to present chains of relations to the user. However, the fingerprint graphs produced by the SSDM computation record only when and where the walkers converged, discarding all information about the path taken by the walkers; furthermore, discarding this information is essential to the calculation as a whole to maintain acceptable space and performance characteristics. While simply listing convergence points does provide the user with *some* intelligible information, it does not provide as rich an explanation as we would like.

Therefore, our approach to explanation generation is to rerun the SSDM calculation on a smaller scale at query-time, and explicitly store the relations traversed rather than condensing the results into a terse fingerprint graph. Two alterations were required to make the modified SSDM calculation efficient enough to provide an acceptable user experience at query time. Both performance improvements were achievable because the modified calculation uses just two walkers rather than starting one walker at every node in the graph. Recall that the similarity of two entities is derived from the expected meeting times of walkers starting at those two entities. If we know in advance which two entities we will be comparing, there is no need to start walkers at any other entities. Because it is so efficient to generate a pair of walks compared to a whole graph's worth of walks, we can afford to run the computation many times per explanation request, and then choose from among the possible explanations to display relevant results to users.

The first performance improvement strategy relates to the permutation-based convergence strategy described in Section II-A. Each step of the ordinary SSDM calculation begins by randomly shuffling all edges in the graph. In the modified calculation, only the two walkers' immediate out-edges need to be shuffled, which is almost always a vastly smaller number of edges, and takes a negligible amount of time.

The second performance improvement strategy relates to the strict edge-label requirements for convergence. In the ordinary SSDM calculation, walkers frequently meet at the same node but do not actually converge because they did not follow the same predicates. In the modified calculation, we instead *require* the two walkers to follow the same predicates as one another. So, in the example given in Fig. 3, path B would never be generated; instead, either Walker 1 and Walker 2 would both follow a `directed` in-edge, or both would follow a `hasActor` out-edge, or both would follow some other shared edge not shown. If the two walkers were ever unable to follow the same predicate in the same direction, then that run of the computation would end. Coupling the edge options of the two walkers greatly reduces the number of trials that fail to converge, leading to a much more efficient calculation.

As a proof-of-concept for this methodology, we implemented an explanation-generating component of Parliament, an open-source triple store developed and maintained by BBN [11]. In Parliament, users are able to browse and view entities in the knowledge base, explore other triples containing an entity, and view a list of similar entities and their SSDM scores. As part of our work on similarity explanations, we added a "why" button for each score, which users can click to produce an on-demand explanation of that score. The software behind the "why" button interacts with the underlying Jena[1] model of the data in Parliament to walk the graph and produce an explanation using the methodology described in this section. Even for datasets with millions of triples, such as the movie dataset, preliminary findings show that accurate explanations could be produced and displayed to the user within seconds.

In summary, explanations for SSDM scores can be efficiently computed on-demand at query time by re-running a modified version of the SSDM random walker calculation.

VI. Using Salience to Bias Explanations

The final contribution we describe in this paper is a method for incorporating *salience* into explanation generation. Salience is a measure of how rare a fact is in a dataset. Our goal was to produce the most "useful" explanations, which we believed would be the explanations with the *most salient* facts, because salient facts are rare and therefore highly descriptive.

We instead discovered that high-salience explanations often come across as obscure because they can contain extremely rare facts. Similarly, low-salience explanations often come across as obvious because they contain extremely common facts. Nonetheless, just as high-salience explanations often have the upside of being very descriptive, low-salience explanations often have the upside of revealing the broadest and most general similarities rather than obscure trivia. Which flavor of explanation is more "useful" likely depends on the goals of the user and requires more research in an application domain.

In this section we present the definition of salience as applied to facts and explain how we constructed a salience-weighted edge generator so that the explanations generated would contain more or fewer high-salience facts. Note that while the following descriptions will focus on biasing explanations by salience, it can be used to favor facts based on any numerical property of those facts.

A. Fact Salience

Salience is a measure of how rare a fact is in a dataset. A *fact* in this case refers not to a whole statement, but to a statement missing its subject or object; that is to say, structures of the form `subject predicate` *`blank`* or *`blank`* `predicate object` (such as `Spielberg directed` *`blank`* or *`blank`* `directed Gladiator`). Facts with a subject and predicate are called left facts because all the information they retain is on the left side of the statement. Similarly, facts with a predicate and an object are called right facts [12].

We now describe how the salience of a fact is calculated. Consider $o(f)$ to be the number of times a fact is expressed in a set of unique `subject predicate object` triples.

[1] http://openjena.org

Consider also $subj$ to be the number of unique subjects present in those triples, and obj to be the number of unique objects. For left facts, salience is calculated as follows:

$$salience(fact) = \frac{1 - log(o(fact))}{log(obj)} \tag{1}$$

And for right facts, the calculation is as follows:

$$salience(fact) = \frac{1 - log(o(fact))}{log(subj)} \tag{2}$$

The conceptual significance of $subj$ and obj in these equations is to count the number of times each fact could potentially occur, since each left fact could potentially appear with every object in the data set, and each right fact could potentially appear with every subject. Facts that actually *do* appear with almost every available subject or object are extremely common, and are thus not very salient. Facts that are expressed about very few of the available subjects or objects are very rare and therefore highly salient. This intuition, and the resultant calculation, is grounded in the information theoretic concept of relative entropy, discussed in [13].

B. Weighting

The objective of salience weighting is to favor high- or low-salience facts in the generated explanations. This bias can be incorporated into the random permutation that is calculated at the beginning of each time step. In the unweighted explanation calculation, the random walkers choose which edge to take by stepping to the neighbor with the smallest index in this permutation. The original reason for the permutation was to encourage walkers that are near one another to converge, but it can also be modified to add other weights and biases into the explanation-generation process.

In order to encourage walkers' edge choices to favor more salient edges, we would like high-salience edges to be more likely to occur at low indices in the permutation. However, we do not want the distribution of salience in the permutation to be too consistent from trial to trial, otherwise low-salience edges will never reach the top of the rankings, restricting the edge choice of the walkers and severely limiting the breadth of explanations produced.

The permutation as originally described in Fogaras and Rácz ranks *nodes* randomly; however, the salience of a node is not a well-defined concept, and so to enable a salience-weighted permutation algorithm it was necessary to switch to ranking *facts*. This modification was justified for the following reason. Under Fogaras and Rácz's formulation, convergence required only that two walkers meet at the same point, and so to encourage convergence it was enough to encourage walkers to choose the same nodes to step to — hence *nodes* were what was ranked. However, under our formulation, convergence requires that the two walkers also follow the same edge label in the same direction. An edge label plus one endpoint makes up a fact, so to encourage convergence, we are justified in encouraging walkers to choose the same facts to walk along.

The weighted-permutation algorithm we developed works according to the common permutation strategy of assigning a random number to each element to be permuted and then sorting by those numbers. We devised a method to incorporate salience into the generation of those random numbers.

We made two attempts at designing the weighting algorithm. Our first, unsuccessful attempt at a biased algorithm contained an oversight which we corrected in the second version. The first, naive approach worked as follows:

1) Each fact f_i is associated with a nonnegative weight w_i
2) Each fact f_i is assigned a random number r_i between 0 and w_i
3) Elements are ranked in ascending order according to r_i

Using this algorithm, elements with a *low* weight are more likely to get a low number relative to other elements, and are therefore more likely to be ranked first.[2] To use this algorithm to favor salient facts, the weight function used to assign the w_i values could be set to $w_i = 1 - salience(f_i)$, or some other function such as $w_i = 1 - \sqrt{salience(f_i)}$.

When testing this algorithm using $w_i = 1 - salience(f_i)$, a failure mode arose: namely, the very same facts would appear at the top time and time again. We determined that the problem occurred with unique facts because their salience is precisely 1, and their weight was therefore 0. When unique facts were assigned a random number between 0 and their weight they were always assigned 0. This meant that all the unique facts were always first in the list, and so only unique facts were ever generated.

To remedy this problem, we added a parameter to increase the randomness. The modified, successful algorithm works as follows:

1) Each fact f_i is associated with a nonnegative weight w_i
2) Given a parameter b, each fact is assigned a random number r_i between 0 and $b + (1 - b) * w_i$
3) Elements are ranked in ascending order according to r_i

In this way, the parameter b can be tuned to increase or decrease the randomness of the permutation.

The properties of the weight function do not constrain the calculation, so most any weight function could be used. To favor low-salience facts, for example, salience or the square root of salience can be used directly. To favor medium-salience facts, $w_i = salience(f_i) * (1 - salience(f_i))$ could be used. Any number of other functions are possible depending on the desired salience characteristics of the resulting explanation.

C. Relevance of Salience-Weighting

As for whether salient facts are actually more useful, it seems to depend on the intended use case. In some cases, obvious paths are more useful, and in other cases, obscure paths are more useful. If the user knows very little about the area of inquiry he or she is likely to prefer explanations that refer to common and well-understood facts and properties. Conversely, if the user is an expert in the domain, he or she is likely to

[2] "Weight" might be a misnomer here as it implies elements with high weights are favored, but the opposite is the case with this algorithm.

prefer more obscure data. A user with average expertise will probably want only middle-salience explanations.

For example, consider a movie-watcher who has never seen any *Star Wars* movies. He or she may be interested to know about low-salience (i.e. common) facts like these:

```
{ofGenre, Science Fiction}
{hasMusicContributor, John Williams}
{hasActor, James Earl Jones}
```

These facts reference well-known people and broad genres, which could help give a novice a grasp of what relates the *Star Wars* movies to one another. High-salience facts such as the following:

```
{hasProducer, Gary Kurtz}
{hasDirector, Irvin Kershner}
{hasEditor, T. M. Christopher}
```

would be too obscure; an unfamiliar viewer is unlikely to know who, say, Irvin Kershner is, as he is not well-known for directing any other blockbusters. However, a *Star Wars* afficionado who already knows that the *Star Wars* movies are Science Fiction films will find the low-salience facts too obvious. He or she may be interested in knowing about the more unusual details that relate *Star Wars Episode IV: A New Hope* and *Star Wars Episode V: The Empire Strikes Back* to one another, and would be pleased to discover that the relatively-unknown editor T. M. Christopher was involved in the production.

Fortunately, our method enables the salience to be weighted by a custom weight function, so that the user can tune the salience of the resulting explanations to his or her needs. Additionally, because explanations are generated on-demand, the desired obscurity could be specified at query time, which would enable the user to ask for more or less obscure explanations in real time as they explore their data.

VII. Applications

Our work on semantic similarity has been developed with an eye towards a variety of applications, primarily in the military and intelligence domains. In general terms, similarity measures and explanations are very useful tools for analysis of large graph-based datasets.

Consider a simple example. Suppose we are collating information on Libya and we encounter the profiles for the following individuals:

- Muammar al-Gaddafi
- Muammar El-Gadhafi
- Moammar Kadaffi

Despite considerable variation in their spellings, these names all refer to the same Libyan former head of state. In fact, some sources report over one hundred ways to spell this person's name [14] due to ambiguities in the transliteration from Arabic. It would be ideal if we can use additional information to disambiguate the names in order to ascertain that these profiles represent the same person. Using information about the relationships (and actions) of the person from each profile, we could derive similarity scores for each pair of profiles, and merge those profiles as appropriate.

In another setting, suppose we are monitoring the network activities of a group of employees of company X. Each employee belongs to one of four departments: Engineering, Sales, Finance, or Corporate. We can monitor email traffic, access to corporate applications, printers, file repositories and other network activities. Let's focus on George, who is in Sales. We receive alerts that George has been accessing financial software and financial projection data files. Is this activity unusual? Our algorithm could be applied to compare George and the profile of his activities with those of employees at his and other departments. Does George seem to be behaving more like employees in Finance than he was before, or less similarly to his fellow employees at Sales than we might expect? If so, it certainly indicates that George's behavior has changed for some reason, and may warrant investigation.

In essence, our similarity work can be applied to any data expressing relationships between entities. Our algorithm is highly scalable, so it can be applied on very large datasets, and our work on explanations allows the similarity results to be clearly communicated to end users of the data analysis. For these reasons, we believe our work to be highly applicable to a wide variety of military and intelligence tasks.

VIII. Future Work

The future of this work lies in two directions. The first is to perform experiments to assess the improvement of using SSDM as compared to SimRank and other approaches. We would also like to perform user tests to determine the perceived utility of different explanations in a real-world environment. The second direction lies in further research on extensions to the work and on new approaches to enhance it.

Experimentation with SSDM will likely take place in a relevant application domain such as social network analysis (in the intelligence domain) or computer network activity analysis (the cyber domain). Metrics for the meaning of similarity will have to be developed for each domain before the algorithm can be evaluated. This is especially true with user testing, where the experiments need to account for subjectivity and prior domain knowledge. Selecting and vetting the appropriate data sets for automated evaluation is another challenge — graph-based data, which is manually annotated for similarity, does not appear to be very common. One approach we may take is to compare structural similarity generated by SSDM to similarity derived from entity attribute comparison (presence and value of certain attributes). Many well-established algorithms exist in this area to provide a baseline for attribute-based similarity.

The most promising direction of further research we envision lies in the domain of calculating *predicate* similarity. With our current algorithm, walkers must traverse identical predicates — a strategy designed to prevent relations such as `A is C` and `B isNever C` from contributing positively to `A`'s similarity with `B`. However, it intuitively seems that `A is C` and `B isOften C` should certainly contribute to `A` and `B`'s similarity. Doing so would require calculating the similarity between predicates (in this case, `is` and `isOften`) before calculating the similarities between entities. One possible way to do this would be to run SSDM on the ontology to calculate predicate similarity before moving on to calculate object similarity. Another possible option would be to use a language-based metric as a source of predicate similarity, such as by using WordNet [15] similarity.

A third option considers the insight that "similar predicates are those that connect similar entities to other similar entities," for example, the predicates `teaches` and `hasInstructor` are considered similar because they appear in relationships such as `Dr. Smith teaches Chemistry` and `CH1301 hasInstructor Dr. Jones` where `Chemistry` is known to be similar to `CH1301` and `Dr. Smith` is similar to `Dr. Jones`. This formulation of predicate similarity is obviously recursive with respect to entity similarity: in order to calculate predicate similarity, we must first calculate entity similarity, and vice versa. However, SimRank, SSDM, and many other algorithms are also based on recursive definitions and derive iterative approximations to the optimal result set. Logically then, we could apply this recursive definition of predicate similarity in order to simultaneously derive both entity (subject and object) and predicate similarity. This is reminiscent of similar iterative and approximate approaches in robotic navigation to solve simultaneous localization and mapping (SLAM) [16][17] problems. It is reasonable to expect that predicate similarity may be derived analogously.

Additional directions also include determining what other weighting schemes besides salience-weighting might be useful. For example, if a dataset includes metadata about provenance or other information about the trustworthiness of each fact (node in the graph), then the SSDM calculation could be weighted to favor more trusted facts over facts from less reliable sources. Other possibilities surely exist.

And finally, while the current explanation format is certainly human-understandable, it does not yet read as easily as text. Fortunately, the data in question is already in subject-predicate-object form; that is to say, it is already in a sentence-like structure, and therefore natural language report generation is a goal well within reach. The additional work required would include determining what sentence structures each predicate fit with, and determining how to ensure that the subjects and objects were tagged with human-readable labels that could be included in a generated report (and not just URIs or other machine-readable descriptors).

IX. Conclusion

Questions of similarity crop up any time users want to make sense of data describing relationships between entities, and data of this form (i.e. graph data or linked data) is ubiquitous. The contributions we describe in this paper help users find similarities in graph data efficiently using SSDM, and understand those similarities using similarity explanations. We defined what an explanation of a similarity score should convey; we implemented a system that can produce such scores and explanations efficiently; and we enabled the obscurity of our explanations to be tuned to meet user's needs. We believe that these contributions will help users in the intelligence and defense communities to make sense of their data, by enabling them to not only find relevant similarities more efficiently, but also to understand those similarities on an intuitive level.

References

[1] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, May 2008. [Online]. Available: http://www.ds3web.it/miscellanea/the_semantic_web.pdf

[2] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data — the story so far," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 5, 2009. [Online]. Available: http://eprints.ecs.soton.ac.uk/21285/1/bizer-heath-berners-lee-ijswis-linked-data.pdf

[3] T. Heath. (2011) Linked data — connect distributed data across the web. [Online]. Available: http://linkeddata.org/

[4] G. Jeh and J. Widom, "SimRank: a measure of structural-context similarity," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '02. New York, NY: ACM, 2002, pp. 538–543. [Online]. Available: http://doi.acm.org/10.1145/775047.775126

[5] M. P. Consens, O. Hassanzadeh, and A. M. Teisanu. (2008, September) Linked movie database. [Online]. Available: http://www.linkedmdb.org/

[6] A. Islam and D. Inkpen, "Semantic text similarity using corpus-based word similarity and string similarity," *ACM Transactions on Knowledge Discovery from Data*, vol. 2, no. 2, pp. 1–25, July 2008. [Online]. Available: http://doi.acm.org/10.1145/1376815.1376819

[7] F. M. Couto, M. J. Silva, and P. M. Coutinho, "Measuring semantic similarity between Gene Ontology terms," *Data and Knowledge Engineering*, vol. 61, no. 1, pp. 137–152, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169023X06000875

[8] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. [Online]. Available: http://books.google.com/books?id=si3R3Pfa98QC

[9] D. Fogaras and B. Rácz, "Scaling link-based similarity search," in *Proceedings of the 14th international conference on World Wide Web*, ser. WWW '05. New York, NY: ACM, 2005, pp. 641–650. [Online]. Available: http://doi.acm.org/10.1145/1060745.1060839

[10] D. Aldous and J. Fill, "Reversible markov chains and random walks on graphs," 2002, in preparation. [Online]. Available: http://stat-www.berkeley.edu/users/aldous/RWG/book.html

[11] 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 (SSWS2009)*, ser. Lecture Notes in Computer Science, vol. 5823. Washington, DC: Springer, October 2009, pp. 17–32. [Online]. Available: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-517/ssws09-paper2.pdf

[12] Commonsense Computing Group. (2011, June) Tutorial: Making matrices from your own data. [Online]. Available: http://csc.media.mit.edu/docs/divisi1/tutorial_make.html

[13] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY: Wiley, 1991. [Online]. Available: http://www.elementsofinformationtheory.com/

[14] S. Bass, "How many different ways can you spell 'Gaddafi'?" *ABC News*, September 2009. [Online]. Available: http://abcnews.go.com/blogs/headlines/2009/09/how-many-different-ways-can-you-spell-gaddafi/

[15] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to wordnet: An on-line lexical database," *International Journal of Lexicography*, vol. 3, no. 4, pp. 235–244, 1990. [Online]. Available: http://wordnet.princeton.edu/

[16] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *The International Journal of Robotics Research*, vol. 5, pp. 56–68, December 1986. [Online]. Available: http://dl.acm.org/citation.cfm?id=33838.33842

[17] J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *IEEE/RSJ International Workshop on Intelligent Robot Systems (IROS)*, November 1991, pp. 1442–1447. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=174711&tag=1

COA modelling with probabilistic ontologies

Henrique C. Marques, José M. P. de Oliveira
Instituto Tecnológico de Aeronáutica
São José dos Campos, BRAZIL
Email: [hmarques, parente]@ita.br

Paulo Cesar G. da Costa
C4I Center - George Mason University
Fairfax, USA
Email: pcosta@c4i.gmu.edu

Abstract—**Planning during complex endeavors is a daunting task in many aspects. An important one is the representation of shared intent, which is an open research topic focused on expressing a common picture among different planning systems with distinct languages, and sometimes disparate problem solving methodologies. The common approach is to use a translator between the order/request message and the planning system, which doesn't convey all the elements that are necessary to support the planning task. The present research proposes to address this issue by the use of a semantic layer as an interface among different planning systems, which not only improves interoperability but also provides support for pruning the search space before the information is sent to the planning system. The layer is based on a probabilistic ontology, which provides shared intent description as well as formalization of the operational domain and of the planning problem, including a principled representation of the involved uncertainty. The proposed scheme supports previous analysis of the search space in order to send to the planning system a concise set of tasks that will contribute to reach the desired end state.**

Keywords—**Interoperability, Automated Planning, Probabilistic Ontology.**

I. Introduction

Complex endeavors are challenging the Command and Control (C2) community with respect to both planning automation and shared intent representation. Both topics are important in order to reach a shared goal during an operation. Because of the collaborative aspect of a joint planning we need to observe the interoperability models in order to provide the level of data representation to be utilised in the planning description.

On the basis of the Organizational Interoperability Maturity Model for C2 (OIMM) [1], the Levels of Conceptual Interoperability Model (LCIM) [2], and the Levels of Information Systems Interoperability (LISI) [3], at least a collaborative level, from an organizational perspective, and a distributed level, from a system perspective, have to be achieved in order to be able to execute a joint planning process [4]. From a data perspective, the semantic (LCIM) interoperability is needed to provide a collaborative (OIMM) - distributed (LISI) level in the highest capability. The semantic interactive level (LCIM Level 3) means that data is shared through the use of a common reference model and content of the information exchange requests is unambiguously defined (see Figure 1).

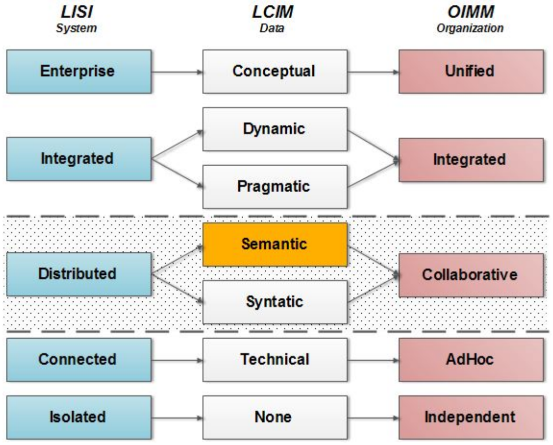


Figure 1. Comparison between interoperability models. Adapted from [4].

Our present research aims to establishing a knowledge representation for improved planning automation that relies on Modeling and Simulation (M&S) interoperability frameworks as its foundational approach. The current major efforts in M&S interoperability are the SISO Coalition-Battle Management Language (C-BML) and the SISO Military Scenario Definition Language (MSDL) [5] [6]. They provide restricted semantic interoperability (mostly relying on the eXtensible Markup Language - XML format) which allows Command and Control systems and simulations to interoperate. One of the reasons behind the restricted semantics is that simulations need less information to generate behavior than what is needed to C2 planning. Since both standards aim to support interoperability among systems and simulations based on the structured XML metadata, their representational demands are comfortably restricted to the smaller information set than what is needed for a C2 planning system. Therefore, Command and Control planning systems cannot take full advantage of the available information until a more expressive approach is used to formaly represent it [7].

The main problem faced by a military planning system is to generate an adequate, feasible, acceptable, and complete plan that is also opportune [8]. In order to support planning automation it is a good practice to represent knowledge in a way that allows for pruning the search space. As a consequence, algorithms ideally have to work with the minimum knowledge that is necessary to produce solutions. This is especially true for the military domain, in which uncertainty is the norm and a plan is usually comprised by a large number of possible tasks whose interaction must reach the desired effects (end state). Also, each organization involved in the operation may have its own planning system, possibly applying a different problem-solving method.

With the development of a more expressive representation to describe the planning domain and the planning problem, it is expected that a planner will have access to more efficient pruning algorithms. This, in turn, will support the identification of solutions for larger problems, as well as to increase the ability to leverage most of the information available to the decision-making process.

Therefore, developing a knowledge representation model and an associated interoperability model are essential steps towards the automation of the planning activity, which is also a major step towards providing alternative Courses of Action (COA) that are reliable, efficient, and opportune. The present research investigates the use of a semantic planning layer, based on a mid-level task probabilistic ontology description as a technical solution for the contextualization of the planning problem. The proposed approach is depicted in Figure 2.

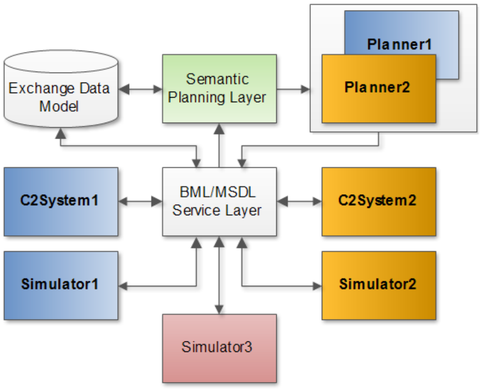


Figure 2. The proposed C2 interoperability framework.

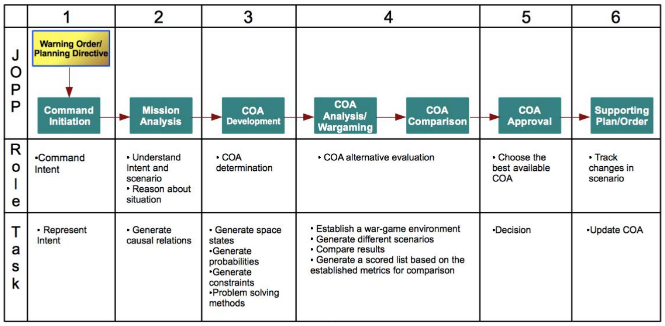
The proposed semantic layer is being developed to support the use of different planning systems in COA development based on a common context description. Section IV describes the layer in more details.

Semantics are essential to align planning automation with a shared intent, while also providing consistency in planning given the orders and requests issued by different organizations.

The paper is divided as follows. Section II provides background on the hierarchical planning process. Section III conveys a brief description of related research addressing automation strategies for operational planning. Section IV addresses the proposed semantic layer, while Section V provides an overview of COA modeling. Section VI describes the COA development based on the adopted methodology, and Section VII concludes this paper with a discussion on the current state of our work.

II. Planning Process

The overall research in this work is grounded on the collaborative aspect of joint planning, and aims to support the Joint Operation Planning Process (JOPP) at the operational level of a joint operation [9]. We chose this process because it involves a joint planning effort within a hierarchical structure with a well established doctrine. Figure 3 shows JOPP from the research development's point of view. The process was divided into six steps, each one with its own role and task to be achieved. The present paper addresses the third step, namely the uncertainty representation during the process of COA determination. For the purpose of this work, the representation of command intent and the description of causal relations will be considered as given. The remaining steps are beyond the scope of this paper.



	1	2	3	4	5	6
JOPP	Warning Order/ Planning Directive Command Initiation	Mission Analysis	COA Development	COA Analysis/ Wargaming COA Comparison	COA Approval	Supporting Plan/Order
Role	•Command Intent	• Understand Intent and scenario • Reason about situation	• COA determination	• COA alternative evaluation	• Choose the best available COA	• Track changes in scenario
Task	• Represent Intent	• Generate causal relations	• Generate space states •Generate probabilities •Generate constraints •Problem solving methods	• Establish a war-game environment • Generate different scenarios • Compare results • Generate a scored list based on the established metrics for comparison	•Decision	•Update COA

Figure 3. The six steps of the Joint Operations Planning Process.

The output of the third step, COA determination, is a representation of a Course of Action with a description of the Measures of Performance (MOP), Measures of Effectiveness (MOE), the planning constraints, and the possible states of the environment.

To produce this output, current decision support systems rely on frameworks that generate orders that are evaluated through simulations. The shared intent is developed via a C2 system GUI that normally generates a set of high level orders and requests that are saved to an exchange data model database. M&S frameworks make use of the SISO Coalition-Battle Management Language (C-BML) [5] message schemata to deliver the command intent, and rely on the SISO Military Scenario Definition Language (MSDL) [6][10] to describe the scenario and the operational domain in terms of spatial situation of allocated resources.

The work in [11] defined an interface between the C2 system's BML output and a standard semantic planning language as the Planning Domain Definition Language (PDDL) [12]. In this scheme, the planning system receives a set of orders converted from the BML format into a more generic planning language, which enables the generation of the right context as a planning problem and a planning domain file.

As a result of the adoption of this scheme, many different planning systems have their own "translator" from BML to a PDDL-like language, usually not aggregating any advantage to the planning process since it does not improve planning agility. In our proposal, we focus on applying ontologies to support automated reasoning over the search space as a means to reduce it before sending the context information to the planning system.

In this approach, the planning system receives only the states, methods and operators that are relevant to the construction of a plan. Efficiency is sought that such this plan can only be generated under the defined constraints and preconditions, and must be in conformance to the desired effect.

III. Related Work

Due to the large spectrum of existing initiatives related to interoperability among command and control (C2) systems, as well as among C2 simulation systems, only those of most interest to this study's context are mentioned here. Initiatives such as the SISO C-BML [13] [14] and MSDL [6] have established the initial structure to support the interoperability among C2 and simulation systems, as well as are setting the standards for addressing the problem of translating the commander's intent into a format that is suitable for simulation and planning systems. The NATO Modeling and Simulation Group Technical Activity 48 (MSG-048) is evaluating a series of technologies to promote such interoperability and is conducting experiments with multinational C2 and simulation systems since 2006 [13] [14].

Another important aspect is to find methods to analyze and evaluate COAs based on effects, as described in [15]. The Effects Based Operations planning significantly increases the number of alternative plans and the depth of evaluation. Therefore, appropriate metrics must be devised to support principled quantification of their relative merits. Generating plans that are aligned with the commander´s intent is a key aspect that may be achieved by the use of semantics during the order generation process. The study conducted in [16] presented results in which all planned orders verified by an ontology-based tool have shown inconsistencies. Such consideration indicates the necessity to utilize semantics in the planning phase to minimize the possibility of inconsistencies with the orders generated at the upper level of the command structure.

In the field of ontology generation for tasking planning, the study in [17] presented an ontology engineering process applicable to such problem. The methodology was straightforward and made explicit the need for breaking down the problem into small pieces, a known strategy in decision theory. The study supports the hypothesis that it is very convenient to manipulate small ontologies that would be integrated later in the process.

Initiatives such as [18] [19] describe the use of task ontologies to support pruning before the planning system receives the planning problem and domain. However, they are not pointing to the interoperability in multilateral application frameworks based on the SISO standards.

Gilmour *et al.* [20] present a solution using a semantic layer in multilateral frameworks to generate plans in accordance with a military ontology. However, the work focus purely in the semantic interoperability of tasks, and does not address the interoperability issue among different planning systems. Thus, in addition to a semantic layer, an ontology extension to support different planning systems has to be established, since each system is likely to have its specific language and a problem-solving method.

The work in [21] is closer to our approach and differs with respect to the implementation and to the ontology integration. While the authors developed a series of military ontologies in OWL language [22], our focus is on achieving interoperability with the reuse of existing ontologies. Another difference is our concern in representing uncertainty in a explicit and principled way, so our approach does address uncertainty representation and reasoning through a mid-level task probabilistic ontology.

IV. Semantic Planning Layer

Different hierarchical levels have to produce a joint operational plan, so different types of planning systems may be utilized throughout the operational campaign. The operational level works with higher level tasks (activities) and is not aware of the exact unity that will handle the task and achieve its desired effect, but it does know which effects will interfere with the desired end-state.

Effects modeling thus play a key role in determining which activities have to be executed in order to achieve the desired effects. It helps improving the tactical level task decomposition by ensuring that only the tasks with higher probabilities to lead to the desired goal effect will be planned at the lower level of the hierarchical chain.

To develop an approach that might handle the effects-based modeling we are proposing a Semantic Planning Layer, which is depicted in Figure 4. As can be seen in the figure, the Semantic Planning Layer is made of a Task Probabilistic Ontology, an Activities Reasoning Module, and a Planning Context Definition Module.

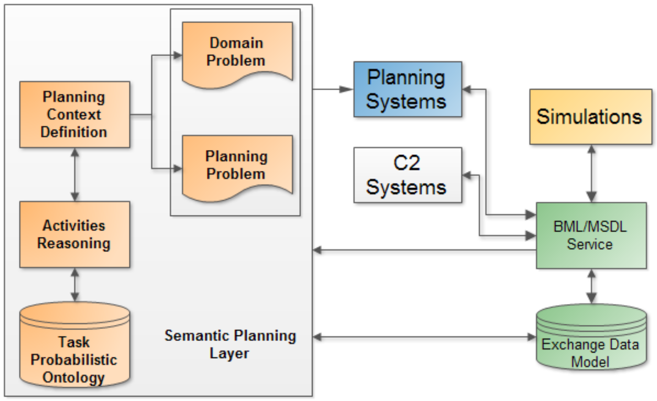


Figure 4. The proposed semantic layer for interoperability between planners and C2 systems.

1) Task Probabilistic Ontology: In order to model the effects and to translate it into a lower level task decomposition, it will be necessary to develop a task ontology that can handle uncertainty. From our perspective, activities are tasks that are more abstract and need to be broken down into smaller tasks until reaching a primitive one. It is also necessary to describe the shared intent in a way that it can be related as desired effects and activities. This is the main reason of our interest in generating a BML ontology description.

Another important description is the domain ontology that will formalize the planning domain specification and interface with other domain descriptions. We are aiming to both describing the hierarchical planning concepts as well as to relate it with the COA description process. The end result will be a better description of the way the activities will be structured in phases and the establishment of a view from the operational perspective.

The mid-level Task Probabilistic Ontology is composed by four ontologies: BML Ontology (BML), Application Domain Ontology (ApplicationDomain), Planning Ontology (HPlanner), and COA Ontology (COA). It is being developed using the PR-OWL probabilistic ontology language [23] and aims to describe the connection between each ontology as well as the causal relations between the main concepts considered during pre-planning reasoning.

The constituent ontologies can be existing ones, which can come from the literature, gold standards, or a particular implementation. The basic premise is that an upper/mid-level ontology describing the core task planning information, and having principled support for uncertainty representation and reasoning will be capable to comprehensively convey all the necessary domain information for planning purposes.

Figure 5 depicts a partial view of the concepts described in the mid-level probabilistic ontology. The hierarchical planner ontology is a specialization of the planning ontology and can be more detailed if needed by a specific problem-solving method. In this scheme, mapping concepts among and between constituent ontologies can be seen as a way of ensuring interoperability from one problem domain to another (*eg.*, from the BML-described commander intent to the Planning domain).

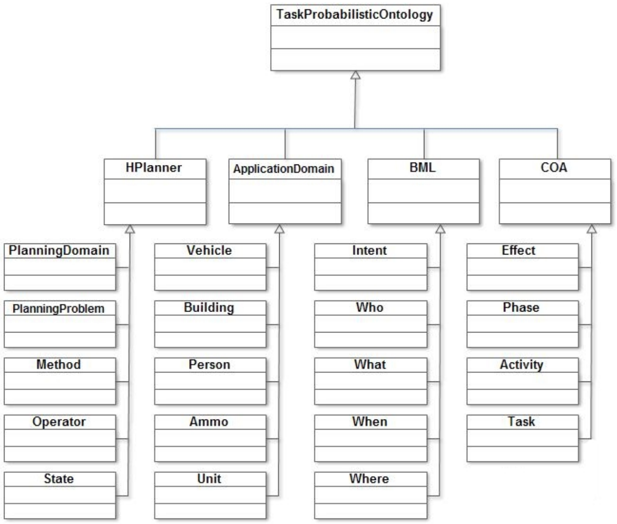
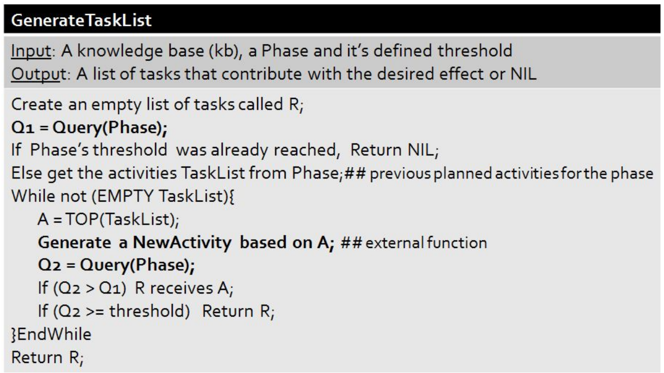


Figure 5. Partial semantic structure of the mid-level Task Probabilistic Ontology.

2) Activities Reasoning: The activities reasoning module executers four main steps:

- Pull BML/MSDL campaign level orders - This step utilizes an already available BML service and no development will be made;
- Identify the activities to be planned through the probabilistic task ontology and by the analyst criteria (defined threshold for each phase (MOE));
- Generate Situation Specific Bayesian Networks (SSBN) [24] to support the activities inference; and
- Export the activities list to be described by the Planning Context definition module;

After a succession of queries, a list with the selected activities will be sent to the Planning Context Definition module. The proposed algorithm is showed below:



```
GenerateTaskList
Input: A knowledge base (kb), a Phase and it's defined threshold
Output: A list of tasks that contribute with the desired effect or NIL
Create an empty list of tasks called R;
Q1 = Query(Phase);
If Phase's threshold was already reached, Return NIL;
Else get the activities TaskList from Phase;## previous planned activities for the phase
While not (EMPTY TaskList){
    A = TOP(TaskList);
    Generate a NewActivity based on A; ## external function
    Q2 = Query(Phase);
    If (Q2 > Q1) R receives A;
    If (Q2 >= threshold) Return R;
}EndWhile
Return R;
```

Figure 6. Pseudo-code for the inference algorithm.

3) Planning Context Definition: The planning context definition is the process of establishing the problem context to be submitted to the planning system. It is composed by three activities:

- Planning Domain definition - After receiving the activities list the module will identify methods that decompose the activities and the operators;
- Planning Problem definition - The planning problem consists of the tasks to be decomposed and the initial state declared on the MSDL message; and
- PDDL files generation.

After receiving the task list, the module has to describe the tasks with the constraints, the current state, and the proposed goal. Such description will then be translated into a PDDL-like format. Finally, the resulting files will be submitted to a domain-independent planning system that will address the planning problem. As depicted in Figure 4, the output are the two PDDL formatted files describing the Domain Problem and the Planning Problem.

V. COA MODELING

Military operations are generally described by phases and activities at the operational level, which are then translated into tasks at the tactical level. The development of Courses of Action follows a decomposition model in the Effects-Based Operations (EBO) paradigm [25]. The modeling effort aims to express a cause-effect relationship from the perspective of activities that will produce outcomes.

Figure 7 shows an example of a phase decomposed into activities and tasks. The arrangement of both the activities and the tasks may be serial, parallel or a combination of both. The task decomposition is a process used in hierarchical planning systems [26] [27]. In our approach, different hierarchical planning systems can receive shared intents and generate different plans that adhere to a mid-level ontology, based on their own problem-solving methods. Hierarchical planning systems were selected because they build plans by hierarchical decomposition that correspond to task models of human task performers. In that way, the generated plans will meet with human approval [28].

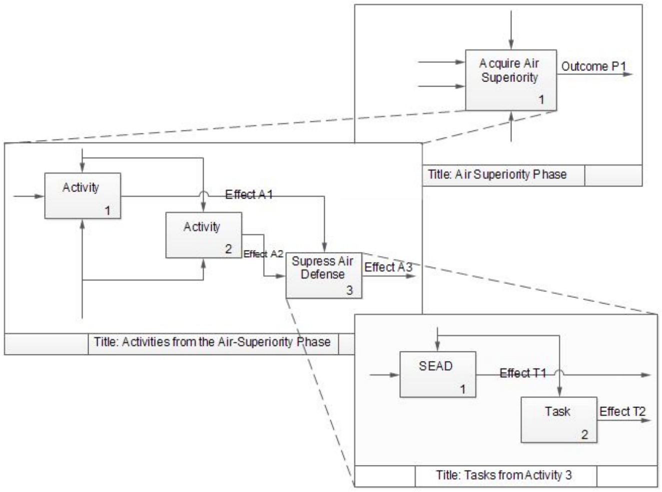


Figure 7. The phase decomposition description in IDEF0 format.

So, in our modeling we describe the COA in terms of phases, activities, tasks, and effects. Figure 8 shows the cumulative effects model we are using to generate queries about the planned tasks. Before sending activities to the planning effort, it is possible to identify the ones that are most important to reach the desired phase's outcome.

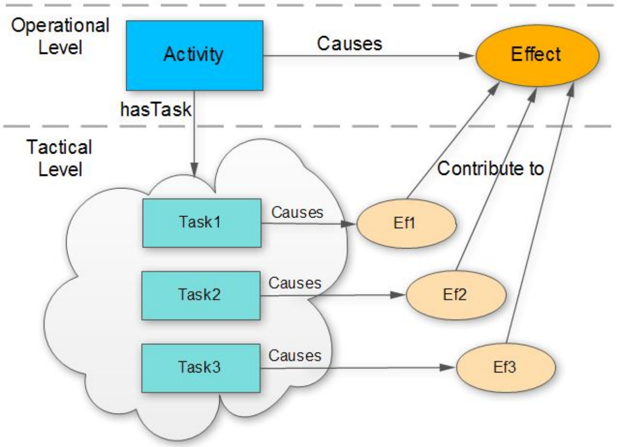


Figure 8. The cumulative effects model.

The process of COA modeling demands a comprehensive method to develop the different ontologies to be utilized in the semantic layer. Our approach relies on ontologies for describing and updating the necessary information to support a planning cell from a military organization in acquiring and maintaining a high-level situational awareness. This requires a formal representation of concepts about time, space, actions, effects, resources, and uncertainty over a dynamic future.

Traditional ontologies do not have built-in mechanisms for representing or inferring with uncertainty, requiring extensions with new classes, subclasses, and properties that support uncertainty representation and reasoning. The PR-OWL probabilistic ontology language [23] and its newest version PR-OWL 2 [29] are written in OWL [22] and provide a consistent framework for representing and reasoning in domains with uncertainty.

The mathematical basis for PR-OWL is Multi-Entity Bayesian Networks - MEBN, which integrates first order logic with Bayesian probability. MEBN provides adequate formal support for representing a joint probability distribution over situations involving unbounded numbers of entities interacting in complex ways [24]. This is a major requirement to achieve principled representation of the multiple, multi-modal sensor input and their compounded interactions. MEBN represents domain information as a collection of inter-related entities and their respective attributes. Knowledge about attributes of entities and their relationships is represented as a collection of repeatable patterns, known as MEBN Fragments (MFrags).

A set of MFrags that collectively satisfies first-order logical constraints ensuring a unique joint probability distribution is a MEBN Theory (MTheory). As in any Bayesian approach, a MEBN model includes the a priori knowledge stored in local probability distributions. The inference process is triggered by one or more queries, which trigger a reasoner that applies Bayesian inference to calculate the marginal distributions.

During a campaign, as new information accrues, this process is used to calculate the posterior probabilities that represent the best knowledge possible to support new planned actions given the information available at the decision time.

VI. COA Development

The COA development starts with the analysis of the activities to be delineated as tasks to the tactical level. Thus, it is necessary to have the operational description of the outcomes in order to reason about the associated likelihoods of reaching the desired effects.

In our model we describe the phases, activities, and effects that will produce the desired end state in a backwards description of the plan. That is, from the desired effect back to the task to be executed as seen in Table I.

The information received from the operational level establishes the COA description and the Domain description. The Domain ontology captures all the information regarding the physical aspects of the operation, and will be utilized to describe the scenario situation. The Effects, Activities and Tasks are described as individuals in our COA ontology (see Figure 9).

TABLE I
EFFECTS TO TASKS.

Phase - Air Superiority		
Outcome - Acquire at least 60% of Air Superiority		
Effect	**Activity**	**Task**
Destroy AAA	SEAD	SEAD
Destroy Radar	Attack Radar	Attack DMPI01 and DMPI02
Destroy C2 Comm	Attack C2 Comm	Attack DMPI03 and DMPI04

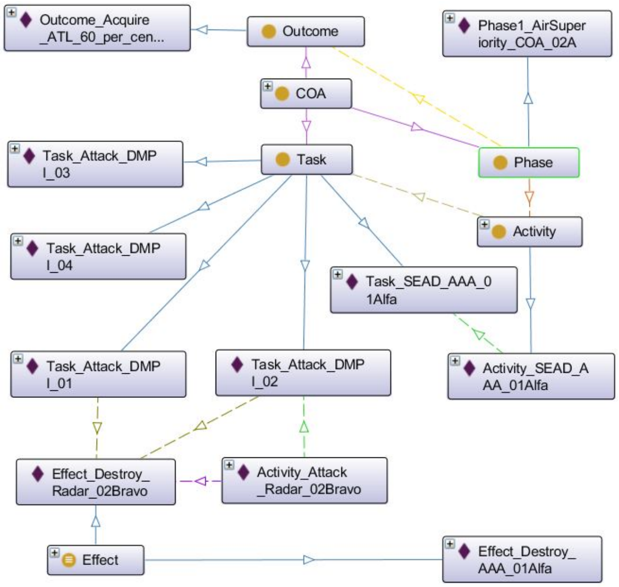

Outcome_Acquire _ATL_60_per_cen...
Outcome
Phase1_AirSuper iority_COA_02A
COA
Task_Attack_DMP I_03
Task
Phase
Activity
Task_Attack_DMP I_04
Task_SEAD_AAA_0 1Alfa
Task_Attack_DMP I_01
Task_Attack_DMP I_02
Activity_SEAD_A AA_01Alfa
Effect_Destroy_ Radar_02Bravo
Activity_Attack _Radar_02Bravo
Effect
Effect_Destroy_ AAA_01Alfa

Figure 9. COA Ontology with individuals exemplifying Table I description.

During the ontology construction we can use the modeling depicted in Figure 8, showing the cumulative effects to support the phase's outcome reasoning. This part of the ontology can be modelled through the probabilistic representation available in PR-OWL. Basically, we model the causal relations in the same way a depicted in Figure 8, establishing a joint probability distribution that will allow reasoning on the available information regarding the current operation situation.

Figure 10 shows a MEBN fragment with only the effects portion of the ontology. The MFrag shows the structure, but not the individuals in the knowledge base. Resident nodes (yellow ovals in the figure) are the actual random variables that form the core subject of the MFrag. Context nodes (green pentagons in the figure) are boolean random variables representing conditions that must be satisfied to make the probability distribution of an MFrag valid. The reasoning occurs by executing a query to support the analysis during the tactical COA development. Thus, given a new set of effects to be reached, one can query the knowledge base for which task might have the greatest influence on a specific effect.

Using the data in Table I we can identify the impact from the Air Superiority phase on the accumulated effect. This takes into account the change in the quantity of a given task from a specific activity. We have modeled the knowledge base with two scenarios:

- One task as the attack in the C2 Comm (*Attack_C2Comm_03Bravo*), two tasks as SEAD missions (*SEAD_AAA_01Alfa*), and one task as the attack in the Radar site (*Attack_Radar_02Bravo*). See Figure 11; and
- Two tasks as the attack in the C2 Comm (*Attack_C2Comm_03Bravo*), two tasks as SEAD missions (*SEAD_AAA_01Alfa*), and one task as the attack in the Radar site (*Attack_Radar_02Bravo*). See Figure 12.

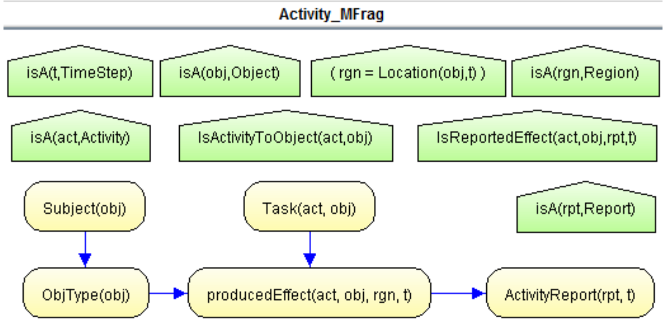


Figure 10. The Activity MFrag depicts the produced effects by a task.

In performing this analysis, one can assess the impact of another attack mission over the C2 Communications facilities with an expected increasing in the accumulated effect by 3.18%. This analysis capability allows for not only to decomposing the activities into tasks as expected for a planning algorithm, but also to identify the activities to be decomposed that will support the expected effect for each phase of the campaign. In the example, the answered query *?hasAccomplishedPhaseGoal* (*Phase1_Air_superiority_COA_02A*) has not reached yet the 60% level defined threshold and other activities will be selected in order to generate the minimum expected outcome for the desired effect based on the model.

VII. CONCLUSION AND FUTURE WORK

The present work involves using a probabilistic ontology language (PR-OWL) to support task analysis and to provide a mid-level ontology as part of a layer between the intent description and the planning system that has to generate a Course of Action. Our approach aims to establish a knowledge representation layer to facilitate pruning the search space. It also verifies the activities that have to be sent to the planner in order to generate the plan that will contribute to reach the desired end state of the campaign.

As future work we have identified the need of improving the effects model to also show the secondary effects produced by the primary effects caused by activities. We also intend to fully implement the semantic layer and to integrate a planning system that is capable to take advantage of the approach. Finally, we plan to test and evaluate our results via a simulation testbed, which is current in development in a shared effort between the GMU C4I Center and the Brazilian Instituto Tecnológico de Aeronáutica.

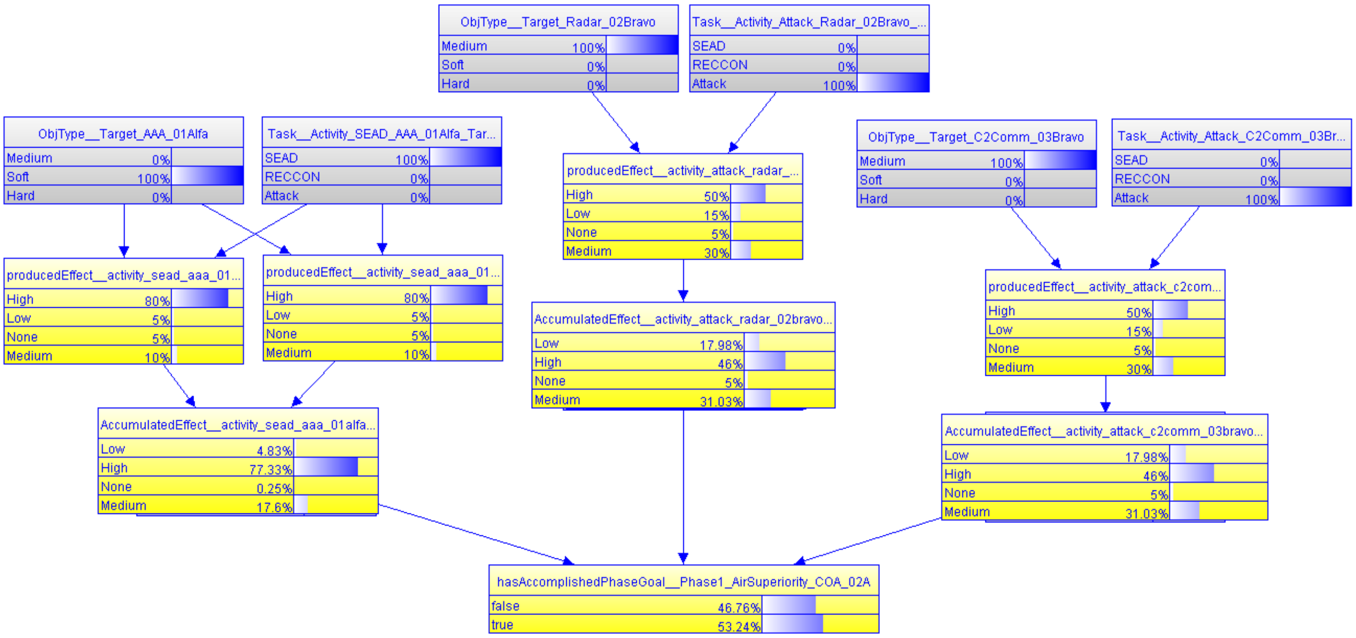

ObjType__Target_Radar_02Bravo
Medium 100%
Soft 0%
Hard 0%
Task__Activity_Attack_Radar_02Bravo_...
SEAD 0%
RECCON 0%
Attack 100%
ObjType__Target_AAA_01Alfa
Medium 0%
Soft 100%
Hard 0%
Task__Activity_SEAD_AAA_01Alfa_Tar...
SEAD 100%
RECCON 0%
Attack 0%
producedEffect__activity_attack_radar_...
High 50%
Low 15%
None 5%
Medium 30%
ObjType__Target_C2Comm_03Bravo
Medium 100%
Soft 0%
Hard 0%
Task__Activity_Attack_C2Comm_03Br...
SEAD 0%
RECCON 0%
Attack 100%
producedEffect__activity_sead_aaa_01...
High 80%
Low 5%
None 5%
Medium 10%
producedEffect__activity_sead_aaa_01...
High 80%
Low 5%
None 5%
Medium 10%
AccumulatedEffect__activity_attack_radar_02bravo...
Low 17.98%
High 46%
None 5%
Medium 31.03%
producedEffect__activity_attack_c2com...
High 50%
Low 15%
None 5%
Medium 30%
AccumulatedEffect__activity_sead_aaa_01alfa...
Low 4.83%
High 77.33%
None 0.25%
Medium 17.6%
AccumulatedEffect__activity_attack_c2comm_03bravo...
Low 17.98%
High 46%
None 5%
Medium 31.03%
hasAccomplishedPhaseGoal__Phase1_AirSuperiority_COA_02A
false 46.76%
true 53.24%

Figure 11. The SSBN of the first scenario. The cumulative effect is 53.24%.

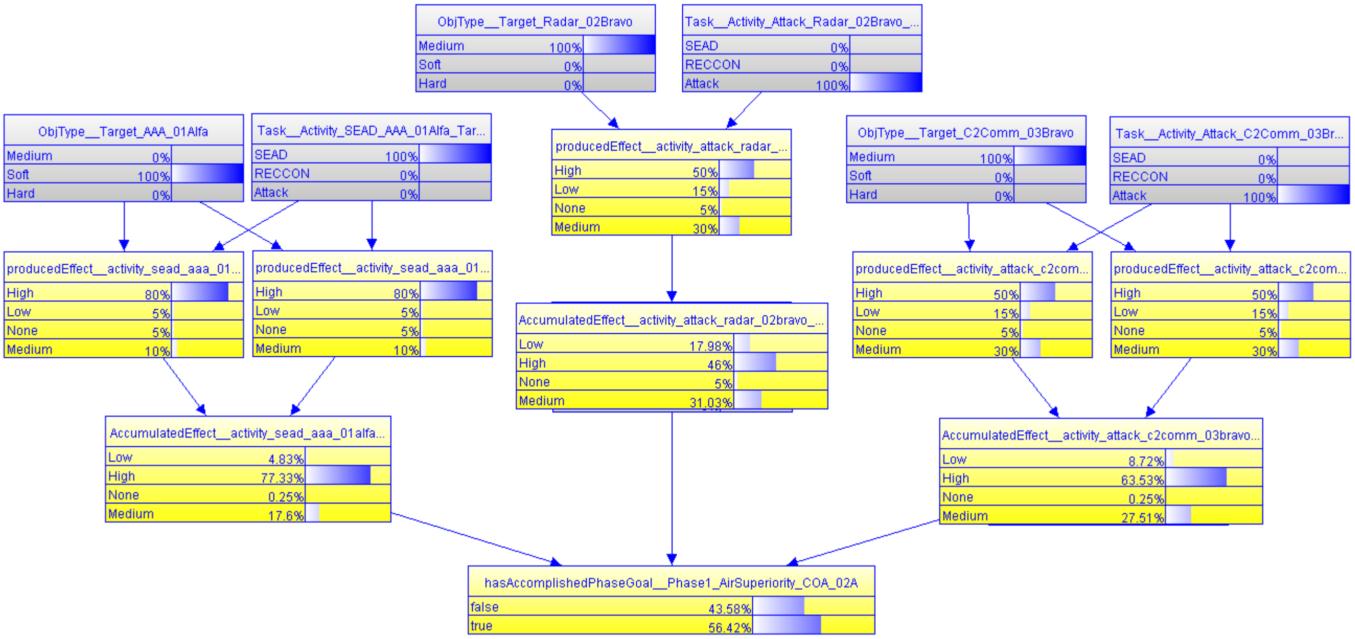

ObjType__Target_Radar_02Bravo
Medium 100%
Soft 0%
Hard 0%
Task__Activity_Attack_Radar_02Bravo_...
SEAD 0%
RECCON 0%
Attack 100%
ObjType__Target_AAA_01Alfa
Medium 0%
Soft 100%
Hard 0%
Task__Activity_SEAD_AAA_01Alfa_Tar...
SEAD 100%
RECCON 0%
Attack 0%
producedEffect__activity_attack_radar_...
High 50%
Low 15%
None 5%
Medium 30%
ObjType__Target_C2Comm_03Bravo
Medium 100%
Soft 0%
Hard 0%
Task__Activity_Attack_C2Comm_03Br...
SEAD 0%
RECCON 0%
Attack 100%
producedEffect__activity_sead_aaa_01...
High 80%
Low 5%
None 5%
Medium 10%
producedEffect__activity_sead_aaa_01...
High 80%
Low 5%
None 5%
Medium 10%
AccumulatedEffect__activity_attack_radar_02bravo_...
Low 17.98%
High 46%
None 5%
Medium 31.03%
producedEffect__activity_attack_c2com...
High 50%
Low 15%
None 5%
Medium 30%
producedEffect__activity_attack_c2com...
High 50%
Low 15%
None 5%
Medium 30%
AccumulatedEffect__activity_sead_aaa_01alfa...
Low 4.83%
High 77.33%
None 0.25%
Medium 17.6%
AccumulatedEffect__activity_attack_c2comm_03bravo...
Low 8.72%
High 63.53%
None 0.25%
Medium 27.51%
hasAccomplishedPhaseGoal__Phase1_AirSuperiority_COA_02A
false 43.58%
true 56.42%

Figure 12. The SSBN of the second scenario. The cumulative effect is 56.42%

References

[1] T. Clark. and R. Jones, "Organisational interoperability maturity model for c2," in *Proceedings of the 1999 Command and Control Research and Technology Symposium.*, United States Naval War College. Newport: ommand and Control Research Program (CCRP), June 1999.

[2] A. Tolk and J. A. Muguira, "The Levels of Conceptual Interoperability Model (LCIM)," in *Proceedings IEEE Fall Simulation Interoperability Workshop*. IEEE CS Press, 2003.

[3] D. o. D. C4ISR Interoperability Working Group, "Levels of information systems interoperability," C4ISR Architectures Working Group, March 1998. [Online]. Available: http://www.c3i.osd.mil/org/cio/i3/

[4] L. S. Winters, M. M. Gorman, and A. Tolk, "Next generation data interoperability: It's all about the metadata," Fall Simulation Interoperability Workshop, Fall Simulation Interoperability Workshop, September 2006, orlando, FL.

[5] S. I. S. Organization, *C-BML- PDG - Coalition-Battle Management Language*, Simulation Interoperability Standards Organization (SISO) Std., April 2011.

[6] ——, *SISO-REF-015-2006 Military Scenario Definition Language (MSDL)*, Simulation Interoperability Standards Organization (SISO) Std., September 2006.

[7] A. Tolk and C. L. Blais, "Taxonomies, ontologies, and battle management languages - recomendations for the coalition bml study group," Spring Simulation Interoperability Workshop, Spring Simulation Interoperability Workshop, April 2005, san Diego.

[8] DoD, *Joint Operation Planning*, joint publication 5 ed., Joint Publication Library, USA, august 2011. [Online]. Available: http://www.dtic.mil/doctrine/new_pubs/jp5_0.pdf

[9] J. Publication, *Command and Control of Joint Air Operations*, January 2010. [Online]. Available: http://www.dtic.mil/doctrine/new_pubs/jp3_30.pdf

[10] (2011, September) Simulation interoperability standards organization. SISO. [Online]. Available: http://www.sisostds.org/Home.aspx

[11] M. Nazih and U. Schade, "Einsatz der battle management language zur befehligung von einheiten in einem simulationssystem," FKIE-Bericht, Tech. Rep. 174, 2009, wachtberg:FGAN.

[12] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins, "PDDL – the planning domain definition language – version 1.2," Yale Center for Computational Vision and Control, Tech. Rep., October 1998.

[13] J. Pullen, M. Hike, S. Levine, A. Tolk, and C. Blais, "Joint battle management language (jbml) - us contribution to the c-bml pdf and nato msg-048 ta," presented at the IEEE European Simulation Interoperability Workshop, June 2007.

[14] J. Pullen, D. Corner, N. Cordonnier, M. Mennane, L. Khimeche, O. M. Mevassvik, A. Alstad, U. Schade, M. Frey, N. d. Reus, P. d. Krom, N. LeGrand, and A. Brook, "Adding reports to coalition battle management language for nato msg-048," in *Joint SISO/SCS European Multiconference*, Instambul, Turkey, July 2009.

[15] C. Egan and J. Reaper, "Course of action scoring and analysis," presented at the 12th International Command and Control Research and Technology Symposium, DOD CCRP. Newport, RI, USA: CCRP Publications, June 2007.

[16] D. Gilmour and Z. M. Zhang, "Determining course of action alignment with operational objectives," presented at the Command and Control Research and Technology Symposium: The state of the art of the practice, June 2006.

[17] A. Frantz and M. Franco, "A semantic web application for the air tasking order," presented at the 10th International Command and Control Research and Technology Symposium: The Future of C2, June 2005.

[18] B. Chandrasekaran, J. R. Josephson, and V. R. Benjamins, "Ontology of tasks and methods," 1998.

[19] A. F. Martins and R. D. A. Falbo, "Models for representing task ontologies."

[20] D. Gilmour, L. Krause, L. Lehman, B. McKeever, and T. Stirtzinger, "Scenario generation to support mission planning," 2006 Command and Control Research and Technology Symposium, 2006.

[21] I.-C. Hsu, Y. K. Tzeng, Y. J. Cheng, and D.-C. Huang, "Semantic-based military scenario generation for mission planning," *Journal of Convergence Information technology*, vol. 6, no. 4, pp. 123–134, April 2011.

[22] W3C. (2004, February) Owl web ontology language. W3C Recommendation. [Online]. Available: http://www.w3.org/TR/owl-features/

[23] P. C. G. da Costa, "Bayesian semantics for the semantic web," Ph.D. dissertation, George Mason University, 2005.

[24] K. B. Laskey. (2007, December) Mebn: A language for first-order bayesian knowledge bases. Online. [Online]. Available: http://ite.gmu.edu/∼klaskey/papers/Laskey_MEBN_Logic.pdf

[25] E. A. Smith, *Effects Based Operations: Applying Network Centric Warfare in Peace, Crisis and War*, DoD, Ed. CCRP, 2002.

[26] Q. Yang, *Intelligent planning: a decomposition and abstraction based approach*. London, UK: Springer-Verlag, 1997.

[27] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning Theory and Practice*. Amsterdam: Elsevier/Morgan Kaufmann, 2004.

[28] R. P. Goldman, "Ontologies and planners a statement of interest," 2004.

[29] R. Carvalho, "Probabilistic ontology: Representation and modeling methodology," Ph.D. dissertation, George Mason University, 2011.

Computational Theory and Cognitive Assistant for Intelligence Analysis

Gheorghe Tecuci, Dorin Marcu, Mihai Boicu, David Schum, Katherine Russell
Learning Agents Center, Volgenau School of Engineering, George Mason University, Fairfax, VA 22032

***Abstract*— This paper presents elements of a computational theory of intelligence analysis and its implementation in a cognitive assistant. Following the framework of the scientific method, this theory provides computational models for essential analysis tasks: evidence marshaling for hypotheses generation, hypotheses-driven evidence collection, and hypotheses testing through multi-INT fusion. Many of these models have been implemented in a web-based cognitive assistant that not only assists an analyst in coping with the astonishing complexity of intelligence analysis, but it also learns from their joint analysis experience.**

Intelligence analysis, scientific method, cognitive assitant, evidence-based reasoning, mixed-initiative reasoning, discovery, ontology, rules, learning, evidence collection, hypotheses testing

I. INTRODUCTION

The purpose of Intelligence Analysis is to answer questions arising in the decision-making process. Often stunningly complex arguments, involving both *imaginative and critical reasoning*, are necessary in order to establish and defend the *relevance*, the *believability*, and the *inferential force* of evidence with respect to the questions asked. The answers are necessarily probabilistic in nature because evidence is always *incomplete* (we can look for more, if we have time), usually *inconclusive* (it is consistent with the truth of more than one answer), frequently *ambiguous* (we cannot always determine exactly what the evidence is telling us), commonly *dissonant* (some of it favors one answer but other evidence favors other answers), and has various degrees of *believability* shy of perfection [1, 2]. Not only is this process highly complex, but it often needs to be performed in a very short period of time.

Given these characteristics of intelligence analysis, we believe that it can be best performed through the mixed-initiative integration of human imagination and computer knowledge-based reasoning [3]. To this purpose we are developing a *Computational Theory of Intelligence Analysis* which is grounded in the science of evidence [4], artificial intelligence, logic, and probability. This theory provides computational models for essential analysis tasks: evidence marshaling for hypotheses generation, hypotheses-driven evidence collection, and hypotheses testing through multi-INT fusion. Many of these models have already been implemented in the TIACRITIS web-based cognitive assistant. The first version of TIACRITIS was developed to help intelligence analysts learn critical thinking skills for evidence-based reasoning, through a hands-on approach, based on predefined analysis cases [2, 5]. That version has now been significantly extended with new capabilities that allow intelligence analysts to formulate and analyze their own hypotheses, and also to learn from the performed analyses.

This paper provides an overview of the current status of the computational theory of intelligence analysis, and its implementation in the extended version of TIACRITIS.

II. INTELLIGENCE ANALYSIS AS CEASELESS DISCOVERY OF EVIDENCE, HYPOTHESES, AND ARGUMENTS

Within the framework of the scientific method, we view intelligence analysis as ceaseless discovery of evidence, hypotheses, and arguments in a non-stationary world. It involves a collaborative process of evidence in search of hypotheses, hypotheses in search of evidence, and evidentiary testing of hypotheses (see Fig. 1). Through *abductive reasoning* (which shows that something is *possibly* true) we generate hypotheses from our observations; through *deductive reasoning* (which shows that something is *necessarily* true) we use our hypotheses to generate new lines of inquiry and discover new evidence; and through *inductive reasoning* (which shows that something is *probably* true) we test our hypotheses with the discovered evidence. Therefore, in this paper we will illustrate the discovery of evidence, hypotheses, and arguments with an analysis example, and then we will show how the same analysis is performed with TIACRITIS.

In our analysis example, Mavis, a counterterrorism analyst, reads in today's Washington Post that a canister containing cesium-137 is missing from the warehouse of the Company XYZ in MD (see evidence E at the bottom-left of Fig. 2). The

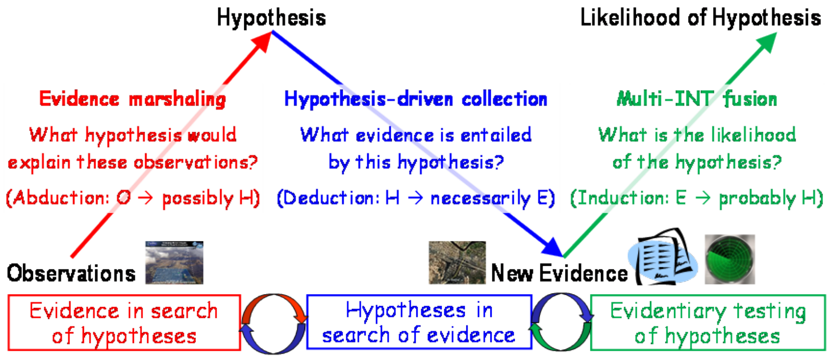

Hypothesis
Likelihood of Hypothesis
Evidence marshaling
What hypothesis would explain these observations?
(Abduction: O → possibly H)
Hypothesis-driven collection
What evidence is entailed by this hypothesis?
(Deduction: H → necessarily E)
Multi-INT fusion
What is the likelihood of the hypothesis?
(Induction: E → probably H)
Observations
New Evidence
Evidence in search of hypotheses
Hypotheses in search of evidence
Evidentiary testing of hypotheses

Figure 1. Framework of the Computational Theory of Intelligence Analysis.

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

question is: *What hypothesis would explain this observation?*

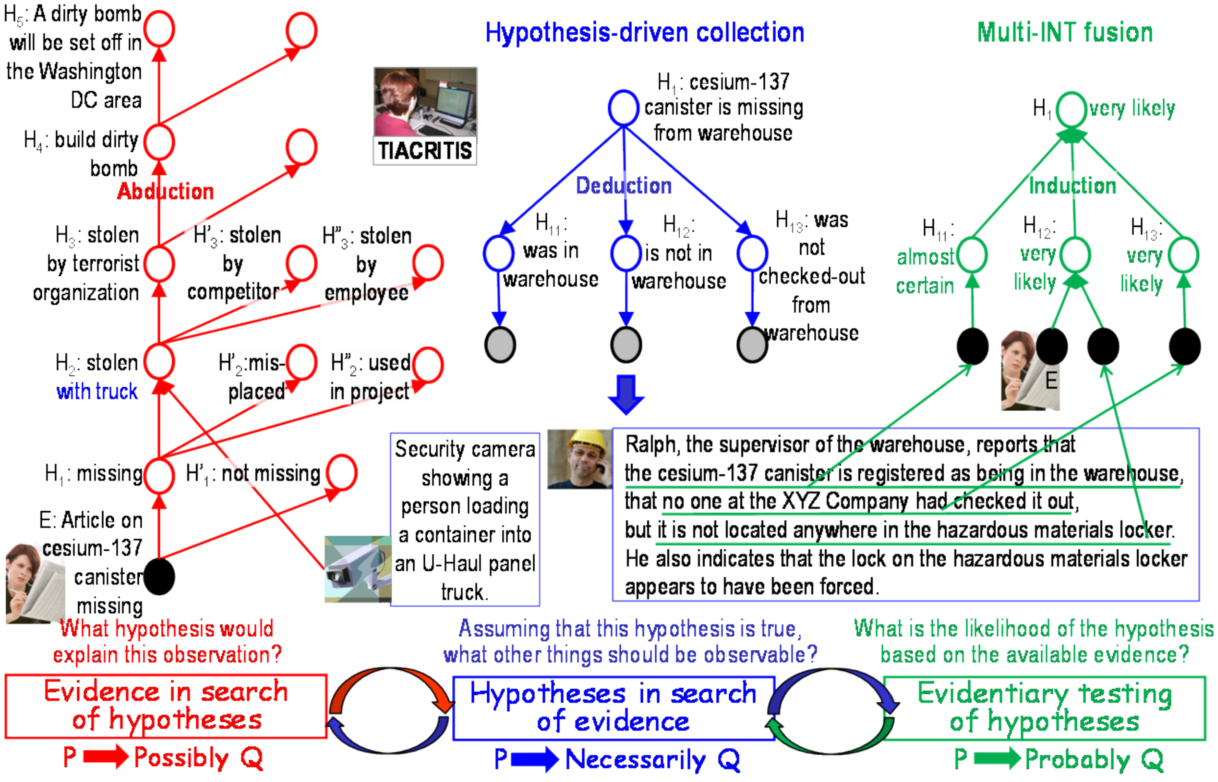
Through *imaginative reasoning*, Mavis *abductively* infers that a dirty bomb will be set off in the Washington, DC area. However, no matter how imaginative or important this hypothesis is, no one will take it seriously unless Mavis and her cognitive assistant, TIACRITIS, are able to justify it. So they develop the chain of abductive inferences shown in the left hand side of Fig. 2. We have evidence that the cesium-137 canister is missing (E). Therefore it is possible that it is indeed missing (H_1). It is possible that it was stolen (H_2). It is possible that it was stolen by someone associated with a terrorist organization (H_3). It is possible that the terrorist organization will use the cesium-137 canister to build a dirty bomb (H_4). It is possible that the dirty bomb will be set off in the Washington, DC area (H_5).

But these are not the only hypotheses that explain E. Just because there is evidence that the cesium-137 canister is missing does not mean that it is indeed missing. At issue here is the believability of the source of this information. Thus an alternative hypothesis is that the cesium-137 canister is not missing (H'_1). But let us assume that it is missing. Then it is possible that it was stolen (H_2). But it is also possible that it was misplaced (H'_2), or maybe it was used in a project at the XYZ Company (H''_2). But let us suppose that it was stolen (H_2). Then it is possible that it was stolen by someone associated with a terrorist organization (H_3). But it is also possible that it was stolen by a competitor (H'_3), or maybe it was stolen by an employee (H''_3), and so on. This is the process of *evidence in search of hypotheses* that would explain it.

The analyst and TIACRITIS need to assess each of these hypotheses before they can conclude that a dirty bomb will be set off in the Washington, DC area. During this process, they would also need to discover who will set off the dirty bomb, and where and when it would be set off.

Starting with H_1, each hypothesis is deductively put to work to guide the collection of additional evidence (see the blue tree in the middle of Fig. 2). Assuming that the cesium-137 canister is indeed missing (H_1), what other things should be observable? Which are the necessary conditions for an object to be reported as missing from a warehouse? It was in the warehouse (H_{11}), it is no longer there (H_{12}), and no one has checked it out (H_{13}). This leads Mavis to contact Ralph, the supervisor of the warehouse, who reports that the cesium-137 canister is registered as being in the warehouse, that no one at the XYZ Company had checked it out, but it is not located anywhere in the hazardous materials locker. He also indicates that the lock on the hazardous materials locker appears to have been forced (see bottom right of Fig. 2). Ralph's testimony provides several items of evidence which are relevant for the hypotheses H_{11}, H_{12}, and H_{13}. This is *hypothesis in search of evidence* that guides the analyst in collecting new evidence.

Mavis and TIACRITIS have now collected more relevant evidence, and the question is: What is the likelihood that the cesium-137 canister is missing, based on the available evidence? To answer this question, they build a Wigmorean probabilistic inference network that shows how the evidence is *fused* through an argument that establishes its relevance, its believability, and its inferential force on the intermediate hypotheses H_{11}, H_{12}, and H_{13} and on the top-level hypothesis H. They conclude that it is very likely the cesium-137 canister


H5: A dirty bomb will be set off in the Washington DC area
H4: build dirty bomb
Abduction
H3: stolen by terrorist organization
H'3: stolen by competitor
H"3: stolen by employee
H2: stolen with truck
H'2:mis-placed
H"2: used in project
H1: missing
H'1: not missing
E: Article on cesium-137 canister missing
TIACRITIS
Hypothesis-driven collection
H1: cesium-137 canister is missing from warehouse
Deduction
H11: was in warehouse
H12: is not in warehouse
H13: was not checked-out from warehouse
Multi-INT fusion
H1 very likely
Induction
H11: almost certain
H12: very likely
H13: very likely
E
What hypothesis would explain this observation?
Assuming that this hypothesis is true, what other things should be observable?
What is the likelihood of the hypothesis based on the available evidence?
Evidence in search of hypotheses
Hypotheses in search of evidence
Evidentiary testing of hypotheses
P ➡ Possibly Q
P ➡ Necessarily Q
P ➡ Probably Q

Security camera showing a person loading a container into an U-Haul panel truck.

Ralph, the supervisor of the warehouse, reports that the cesium-137 canister is registered as being in the warehouse, that no one at the XYZ Company had checked it out, but it is not located anywhere in the hazardous materials locker. He also indicates that the lock on the hazardous materials locker appears to have been forced.

Figure 2. Discovery of evidence, hypotheses, and arguments.

is missing (see the green tree in the right hand side of Fig. 2).

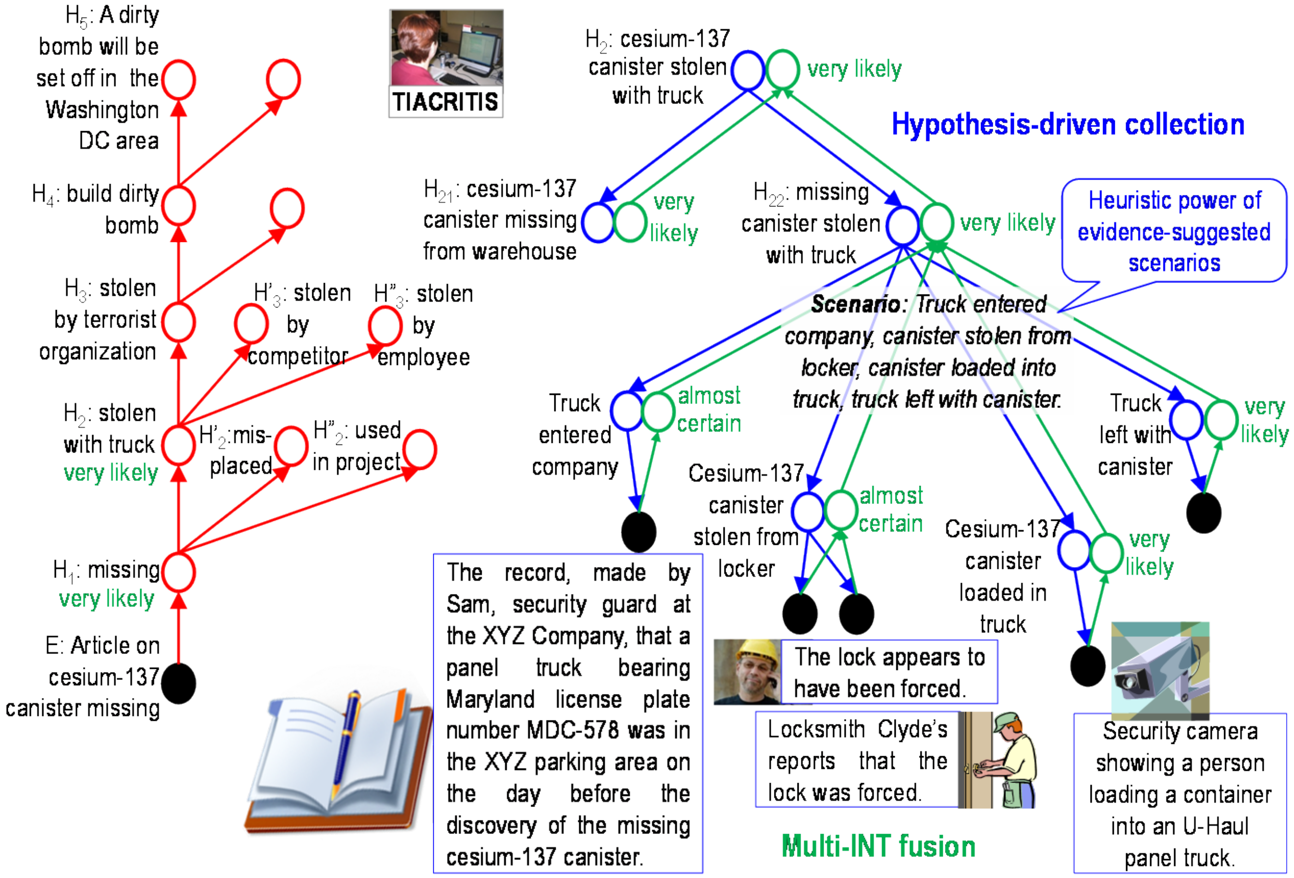
Now, some of the newly discovered items of evidence may trigger new hypotheses, or the refinement of the current hypotheses. Therefore these processes of evidence in search of hypotheses, hypotheses in search of evidence, and evidentiary testing of hypotheses, take place at the same time, and in response to one another, as indicated by the arrows at the bottom of Fig. 2. For example, during her investigation of the security camera of the XYZ warehouse, Mavis discovers a video segment showing a person loading a container into a U-Haul panel truck. Therefore the hypothesis H_2 is refined to "the cesium-137 canister was stolen with the U-Haul panel truck" (see the left part of Fig. 2).

Having concluded that the cesium-137 canister is missing, Mavis and TIACRITIS now have to establish whether the cesium-137 canister was stolen with a truck (H_1), misplaced (H'_1), or used in some project (H''_1). Each of these hypotheses is put to work to guide the collection of relevant evidence which is then used to assess it, as illustrated in Fig. 3.

Assuming that the cesium-137 canister was stolen with a truck (H_2), what other things should be observable? The current evidence suggests the following scenario of how the cesium-137 might been stolen: The truck entered the company, the canister was stolen from the locker, the canister was loaded into the truck, and the truck left with the canister (see the blue tree in the right side of Fig. 3). Such scenarios have enormous heuristic value in advancing the investigation because they consist of mixtures of what is taken to be factual and what is conjectural. Conjecture is necessary in order to fill in natural gaps left by the absence of evidence. Each such conjecture opens up a new avenue of investigation, and the discovery of additional evidence, if the scenario turns out to be true. In this case, for instance, Mavis is led to check whether the truck entered the XYZ parking area. She investigates the record of the security guard and discovers that a panel truck bearing Maryland license plate number MDC-578 was in the XYZ parking area the day before it was discovered that the cesium-137 canister was missing (see the bottom of Fig. 3).

Fusing all the discovered evidence, Mavis and TIACRITIS conclude that it is very likely that the cesium-137 canister was stolen with the MDC-678 truck. However, they now need to also assess H'_2 and H''_2. They do not find any relevant evidence for H'_2. In searching for evidence relevant to H''_2, Mavis contacts Grace, the Vice President for Operations at XYZ. Grace tells Mavis that no one at the XYZ Company had checked the canister out for work on any project. She says that the XYZ Company has other projects involving hazardous materials but none that involves the use of cesium-137. As a result, it is concluded to be very unlikely that the cesium-137 canister was used in a project at the XYZ Company.

Through such *spiral hybrid reasoning*, where abductions, deductions, and inductions feed on each other in recursive calls, Mavis and TIACRITIS continuously generate and update intermediate alternative hypotheses, use these hypotheses to guide the collection of relevant evidence, and use the evidence


H5: A dirty bomb will be set off in the Washington DC area
H4: build dirty bomb
H3: stolen by terrorist organization
H'3: stolen by competitor
H"3: stolen by employee
H2: stolen with truck
very likely
H'2:mis-placed
H"2: used in project
H1: missing
very likely
E: Article on cesium-137 canister missing
TIACRITIS
H2: cesium-137 canister stolen with truck
very likely
Hypothesis-driven collection
H21: cesium-137 canister missing from warehouse
very likely
H22: missing canister stolen with truck
very likely
Heuristic power of evidence-suggested scenarios
Scenario: Truck entered company, canister stolen from locker, canister loaded into truck, truck left with canister.
Truck entered company
almost certain
Cesium-137 canister stolen from locker
almost certain
Cesium-137 canister loaded in truck
very likely
Truck left with canister
very likely
Multi-INT fusion

The record, made by Sam, security guard at the XYZ Company, that a panel truck bearing Maryland license plate number MDC-578 was in the XYZ parking area on the day before the discovery of the missing cesium-137 canister.

The lock appears to have been forced.

Locksmith Clyde's reports that the lock was forced.

Security camera showing a person loading a container into an U-Haul panel truck.

Figure 3. Spiral hybrid reasoning involving synergistic abductive, deductive, and inductive steps.

to test these hypotheses, until the likelihood of the top-level hypothesis is assessed. At the same time, TIACRITIS learns reasoning patterns from the analyst, and becomes increasingly more knowledgeable, as will be illustrated in Section IV.

III. Hypothesis Analysis Through Problem Reduction and Solution Synthesis

The analyst and TIACRITIS analyze hypotheses by employing a general divide and conquer approach, called *problem reduction and solution synthesis*, which combines the deductive and inductive reasoning trees, as shown in the right hand side of Fig. 3. This approach is grounded in the problem reduction representations developed in artificial intelligence [6-8], and in the argument construction methods provided by the noted jurist John H. Wigmore [9], the philosopher of science Stephen Toulmin [10], and the evidence professor David Schum [1]. In this approach, which is illustrated in Fig. 4, the problem of assessing a complex hypothesis H is successively reduced to the assessment of simpler and simpler hypotheses, down to the level of elementary hypotheses. Then these elementary hypotheses (e.g., H_2) are assessed based on the available evidence. Finally, the solutions of these assessments are successively combined, from bottom-up, to obtain the solution of the top level hypothesis assessment.

In Fig. 4 the assessment of the hypothesis H is reduced to the assessment of three simpler hypotheses, H_1, $H_{2,}$ and H_3. The middle hypothesis H_2 is assessed based on the available evidence. As indicated in Fig. 4, one has to consider both *favoring evidence* and *disfavoring evidence*. In this example there are two items of favoring evidence, E_1 and E_2. Therefore one has to assess to what extent each of them favors the hypothesis H_2. This requires the assessment of the *relevance* and *believability* of E_1, and of its *inferential force* on H_2.

The *relevance* answers the question: So what? How does this item of evidence bears on what we are trying to prove or disprove? The *believability* answers the question: Can we believe what this item of evidence is telling it? The *inferential force or weight* answers the question: How strong is this item of relevant evidence in favoring or disfavoring various alternative hypotheses we are entertaining?

As indicated before, all these assessments are probabilistic and, in our research, we have considered symbolic probabilities with names that are similar to those from the US National Intelligence Council's standard estimative language. For example, as shown in the table from the left side of Fig. 4, indicating that a hypothesis is "likely" is equivalent to saying that its probability of being true is between 0.55 and 0.75. Of course, the actual symbolic probabilities and the associated intervals from Fig. 4 are just examples. A user may decide to use other names for symbolic probabilities, as well as other associated intervals, as discussed by Kent [11] and Weiss [12].

In this example let us assume the following solutions for the relevance and the believability of E_1: *"If we believe E_1 then H_2 is almost certain"* and *"It is likely that E_1 is true."* These assessments need to be composed to assess the inferential force of E_1 on H_2. TIACRITIS uses the *"minimum"* composition function, because an item of evidence needs to be both very relevant and very believable to convince us that the hypothesis is true. As a result, the assessed the inferential force of E_1 on H_2 is: *"Based on E_1 it is likely that H_2 is true."* The inferential force of E_2 on H_2 is similarly assessed by TIACRITIS as *almost certain*. Then TIACRITIS composes the inferential force of E_1 on H_2 with the inferential force of E_2 on H_2, by using the *"maximum"* function because it is enough to be convinced by one item of evidence that the hypothesis is true. As a result, TIACRITIS assesses the following inferential force of the favoring evidence (i.e. both E_1 and E_2) on H_2: *"Based on the favoring evidence it is almost certain that H_2 is true."* Through a similar process TIACRITIS assesses the inferential force of the disfavoring evidence on H_2, and then the likelihood of H_2 based on both the favoring and the disfavoring evidence. H_1 and H_3 are assessed in a similar way as *very likely* and *likely*, respectively. Then the assessments of H_1, H_2, and H_3 are combined by TIACRITIS through a function selected by the analyst, such as *minimum* (all three hypotheses required to be true), *maximum* (one hypothesis required to be true), *average*, or *weighted sum*, into the assessment of the top level hypothesis H.

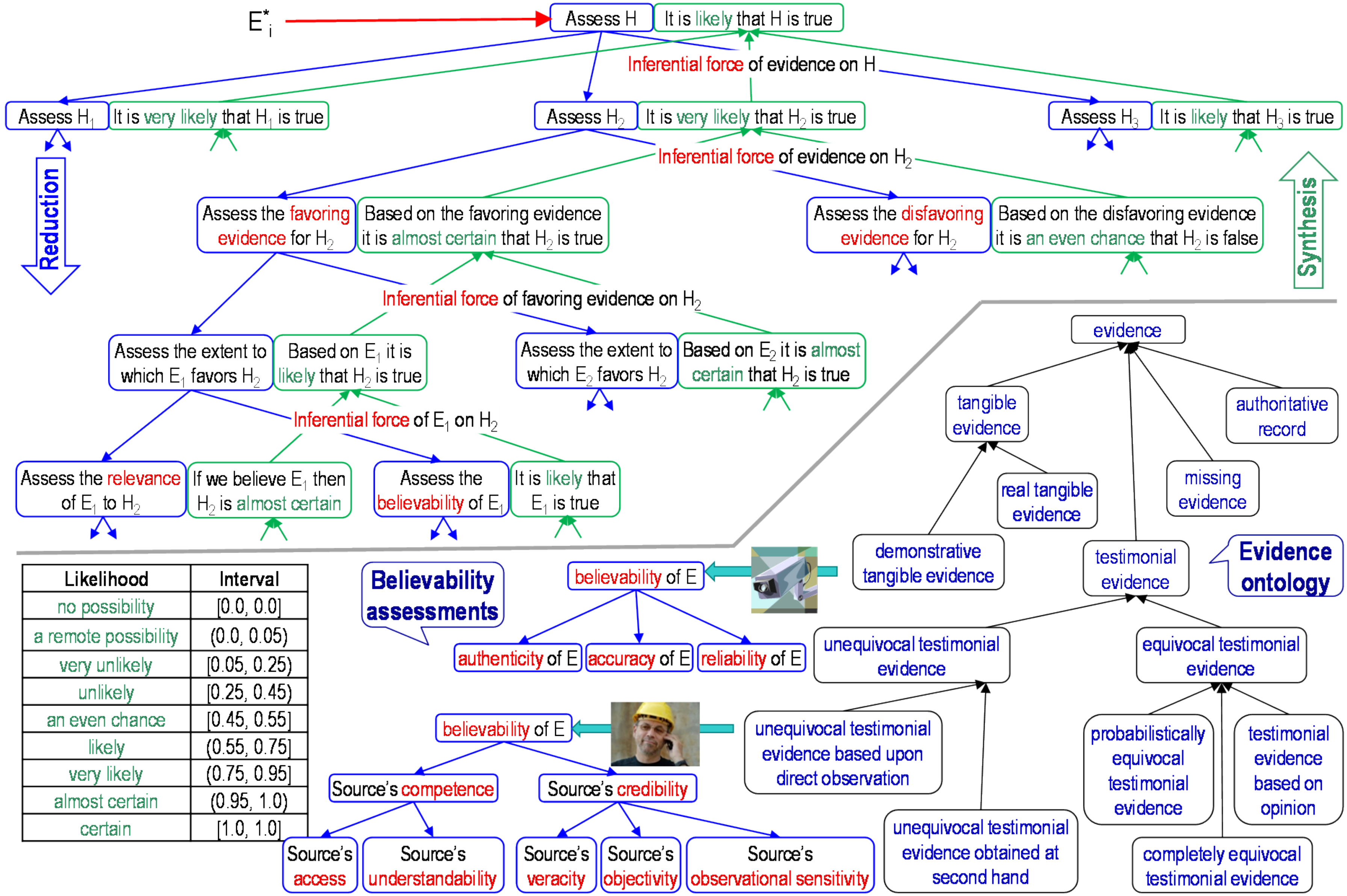
TIACRITIS is able to significantly help the analyst because it has a lot of knowledge about evidence. This includes an *ontology of evidence*, a fragment of which is shown in the bottom-right part of Fig. 4. This ontology distinguishes between different types of *tangible* and *testimonial evidence*. For each such type, TIACRITIS automatically employs a specific believability assessment procedure. For instance, in the case of an item of *demonstrative tangible evidence* which is a representation or image of a tangible thing (e.g., the record of the security camera in Fig. 2), its believability depends on its *authenticity*, *accuracy*, and *reliability*. Also, the believability of *unequivocal testimonial evidence based upon direct observation* (such as Ralph's testimony in Fig. 2) depends on source's *competence* and *credibility*. Competence depends on *access* and *understandability*, while credibility depends on *veracity*, *objectivity*, and *observational sensitivity* [1, 2].

This knowledge allows TIACRITIS to automatically reduce the assessment of complex hypotheses to the assessment of the relevance and believability credentials of evidence, as well as to automatically compose these assessments, once they are made by the analyst.

IV. Illustration of the Use of TIACRITIS

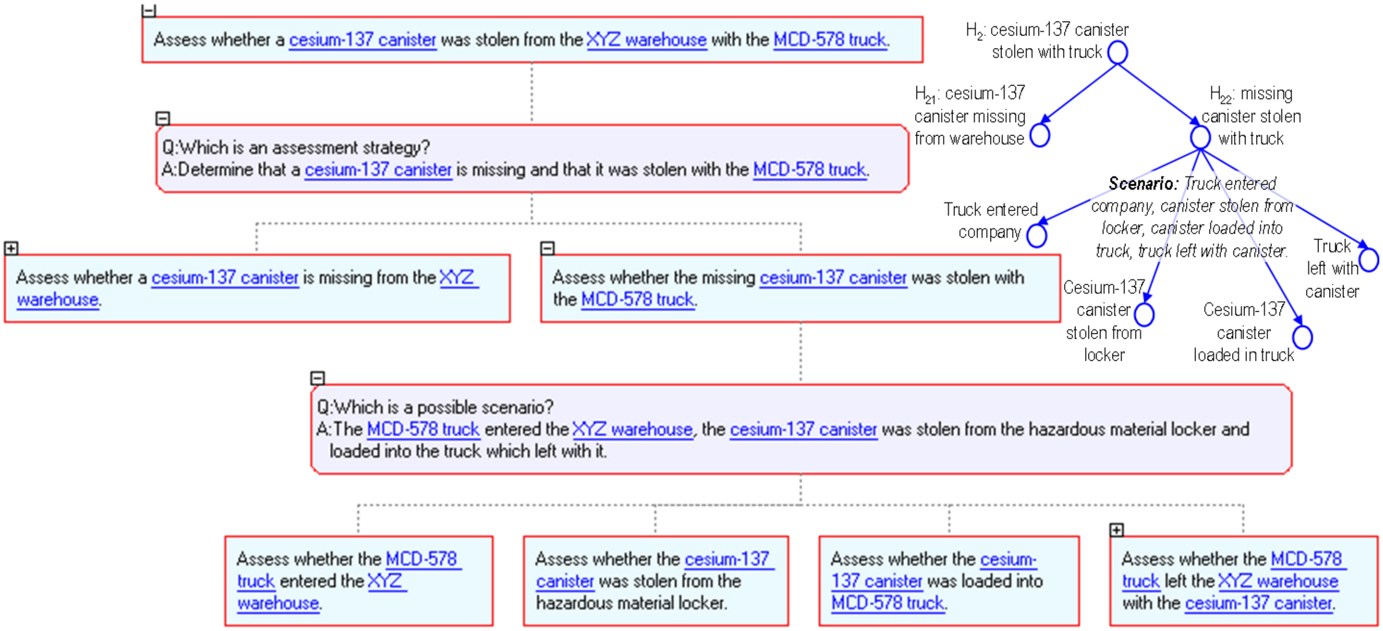
TIACRITIS allows its users to formulate hypotheses, develop argumentation structures to assess them, collect evidence, associate evidence to elementary hypotheses, assess and justify the relevance and the believability of evidence, make assumptions with respect to certain sub-hypotheses, select the composition functions for determining the inferential force of evidence, and assess the hypotheses. We will illustrate these capabilities with the example of assessing the hypothesis H_2 and its argumentation structure from the right side of Fig. 3.

Using TIACRITIS, the analyst formulates the hypothesis analysis problem in English and selects its instances, as shown in the top part of Fig. 5. Selecting the instances allows TIACRITIS to learn the following general hypothesis analysis pattern: "Assess whether a ?O1 was stolen from the ?O2 with the ?O3."


E*i
Assess H
It is likely that H is true
Inferential force of evidence on H
Assess H1
It is very likely that H1 is true
Assess H2
It is very likely that H2 is true
Assess H3
It is likely that H3 is true
Inferential force of evidence on H2
Reduction
Synthesis
Assess the favoring evidence for H2
Based on the favoring evidence it is almost certain that H2 is true
Assess the disfavoring evidence for H2
Based on the disfavoring evidence it is an even chance that H2 is false
Inferential force of favoring evidence on H2
Assess the extent to which E1 favors H2
Based on E1 it is likely that H2 is true
Assess the extent to which E2 favors H2
Based on E2 it is almost certain that H2 is true
Inferential force of E1 on H2
Assess the relevance of E1 to H2
If we believe E1 then H2 is almost certain
Assess the believability of E1
It is likely that E1 is true
evidence
tangible evidence
authoritative record
real tangible evidence
missing evidence
demonstrative tangible evidence
testimonial evidence
Evidence ontology
unequivocal testimonial evidence
equivocal testimonial evidence
unequivocal testimonial evidence based upon direct observation
probabilistically equivocal testimonial evidence
testimonial evidence based on opinion
unequivocal testimonial evidence obtained at second hand
completely equivocal testimonial evidence
Believability assessments
believability of E
authenticity of E
accuracy of E
reliability of E
believability of E
Source's competence
Source's credibility
Source's access
Source's understandability
Source's veracity
Source's objectivity
Source's observational sensitivity

Likelihood	Interval
no possibility	[0.0, 0.0]
a remote possibility	(0.0, 0.05)
very unlikely	[0.05, 0.25)
unlikely	[0.25, 0.45)
an even chance	[0.45, 0.55]
likely	(0.55, 0.75]
very likely	(0.75, 0.95]
almost certain	(0.95, 1.0)
certain	[1.0, 1.0]

Figure 4. Evidence-based hypothesis analysis through reduction and synthesis.



Assess whether a cesium-137 canister was stolen from the XYZ warehouse with the MCD-578 truck.

Q:Which is an assessment strategy?
A:Determine that a cesium-137 canister is missing and that it was stolen with the MCD-578 truck.

Assess whether a cesium-137 canister is missing from the XYZ warehouse.

Assess whether the missing cesium-137 canister was stolen with the MCD-578 truck.

Q:Which is a possible scenario?
A:The MCD-578 truck entered the XYZ warehouse, the cesium-137 canister was stolen from the hazardous material locker and loaded into the truck which left with it.

Assess whether the MCD-578 truck entered the XYZ warehouse.

Assess whether the cesium-137 canister was stolen from the hazardous material locker.

Assess whether the cesium-137 canister was loaded into MCD-578 truck.

Assess whether the MCD-578 truck left the XYZ warehouse with the cesium-137 canister.

H_2: cesium-137 canister stolen with truck
H_{21}: cesium-137 canister missing from warehouse
H_{22}: missing canister stolen with truck
Scenario: Truck entered company, canister stolen from locker, canister loaded into truck, truck left with canister.
Truck entered company
Truck left with canister
Cesium-137 canister stolen from locker
Cesium-137 canister loaded in truck

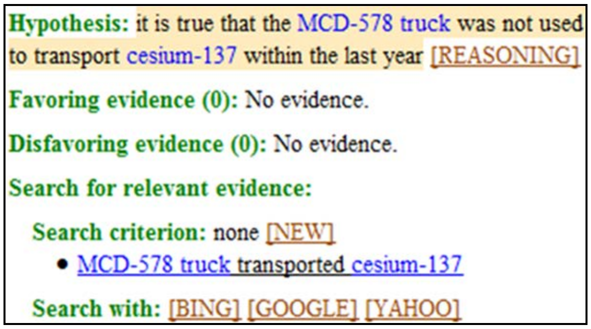
Figure 5. Hypothesis reduction.

As previously described, the analyst and TIACRITIS then reduce this hypothesis analysis problem to simpler and simpler problems, down to the level of elementary hypothesis analysis problems to be solved based on evidence. Notice that each hypothesis analysis problem in Fig. 5 is followed by a question/answer pair which guides its reduction to simpler problems. Thus the top level problem is reduced to two subproblems. The second subproblem is further reduced to four subproblems, based on the scenario discussed in Section II and illustrated in Fig. 3. Some of these reduction steps may be suggested by TIACRITIS, if it has encountered similar steps in past analyses.

Next the analyst will directly assess the elementary hypotheses based on relevant evidence, as discussed below. The analyst may associate any number of search criteria with elementary hypotheses which are then used by TIACRITIS to search for evidence in various repositories, as illustrated in Fig. 6. The top part of this figure shows an elementary hypothesis for which there is no evidence. The bottom part shows a search criteria defined by the analyst, to guide TIACRITIS in searching for relevant evidence on the Internet with BING, GOOGLE, or YAHOO (other search engines and repositories can be added).

The analyst may easily define new items of evidence and may associate them with the hypotheses they favor or disfavor, as illustrated in Fig. 7. The top part of this figure is the description of the evidence item EVD-002-Ralph: Ralph's testimony that the cesium-137 canister is registered as being in the XYZ warehouse. The analyst has selected its type as unequivocal testimonial evidence based upon direct observation. Then the analyst indicated that this item of evidence favors the hypothesis "the cesium-137 canister was in the XYZ warehouse before being reported as missing," as shown in the middle part of Fig. 8.

As a result, TIACRITIS automatically generated the corresponding evidence-based analysis, as shown in Fig. 8. Notice that it considered both favoring and disfavoring



Hypothesis: it is true that the MCD-578 truck was not used to transport cesium-137 within the last year [REASONING]

Favoring evidence (0): No evidence.

Disfavoring evidence (0): No evidence.

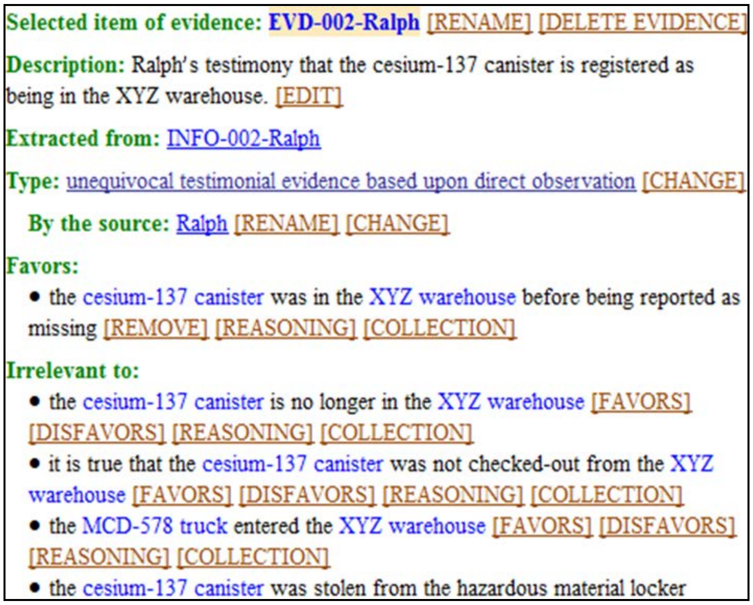
Search for relevant evidence:

Search criterion: none [NEW]

- MCD-578 truck transported cesium-137

Search with: [BING] [GOOGLE] [YAHOO]

Figure 6. Evidence collection.



Selected item of evidence: **EVD-002-Ralph** [RENAME] [DELETE EVIDENCE]

Description: Ralph's testimony that the cesium-137 canister is registered as being in the XYZ warehouse. [EDIT]

Extracted from: INFO-002-Ralph

Type: unequivocal testimonial evidence based upon direct observation [CHANGE]

By the source: Ralph [RENAME] [CHANGE]

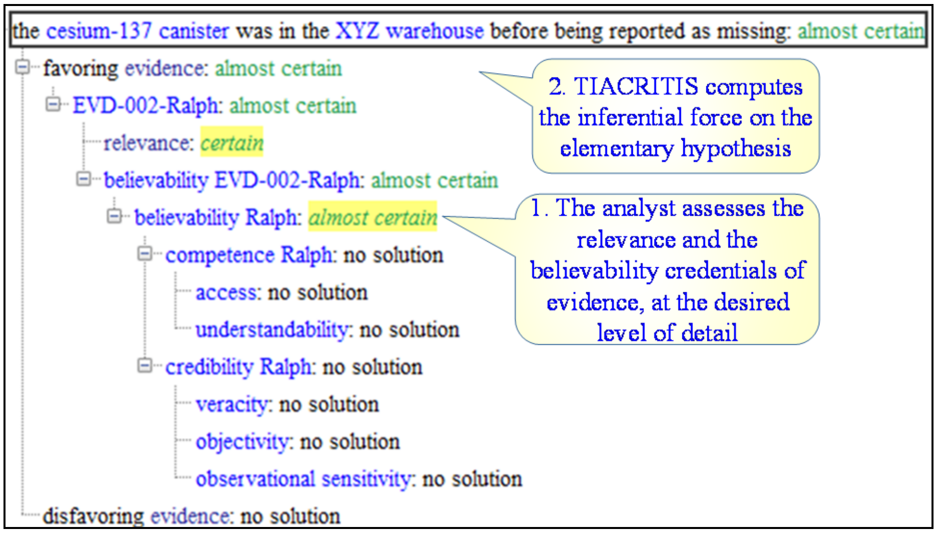
Favors:

- the cesium-137 canister was in the XYZ warehouse before being reported as missing [REMOVE] [REASONING] [COLLECTION]

Irrelevant to:

- the cesium-137 canister is no longer in the XYZ warehouse [FAVORS] [DISFAVORS] [REASONING] [COLLECTION]
- it is true that the cesium-137 canister was not checked-out from the XYZ warehouse [FAVORS] [DISFAVORS] [REASONING] [COLLECTION]
- the MCD-578 truck entered the XYZ warehouse [FAVORS] [DISFAVORS] [REASONING] [COLLECTION]
- the cesium-137 canister was stolen from the hazardous material locker

Figure 7. Evidence representation and use.



the cesium-137 canister was in the XYZ warehouse before being reported as missing: almost certain

- favoring evidence: almost certain
  - EVD-002-Ralph: almost certain
    - relevance: certain
    - believability EVD-002-Ralph: almost certain
      - believability Ralph: almost certain
        - competence Ralph: no solution
          - access: no solution
          - understandability: no solution
        - credibility Ralph: no solution
          - veracity: no solution
          - objectivity: no solution
          - observational sensitivity: no solution
- disfavoring evidence: no solution

2. TIACRITIS computes the inferential force on the elementary hypothesis

1. The analyst assesses the relevance and the believability credentials of evidence, at the desired level of detail

Figure 8. Evidence-based assessment of an elementary hypotheses.

evidence, and included EVD-002-Ralph as favoring evidence for which the analyst needs to assess the relevance and the believability. Because EVD-002-Ralph is unequivocal testimonial evidence based upon direct observation, its believability depends on Ralph's competence and credibility. Competence depends on access and understandability, while credibility depends on veracity, objectivity, and observational sensitivity.

The analyst has assessed the relevance of EVD-002-Ralph as certain and the believability of Ralph as almost certain. Then TIACRITIS has combined these assessments into an inferential force of almost certain, and has computed the likelihood of the corresponding elementary hypothesis.

Notice that although TIACRITIS has provided a detailed believability analysis, the user may drill down into this analysis at the desired level and, in this case, decided to assess directly the believability of Ralph, rather than assessing lower level believability credentials, such as veracity. This is referred to as an *assumption*.

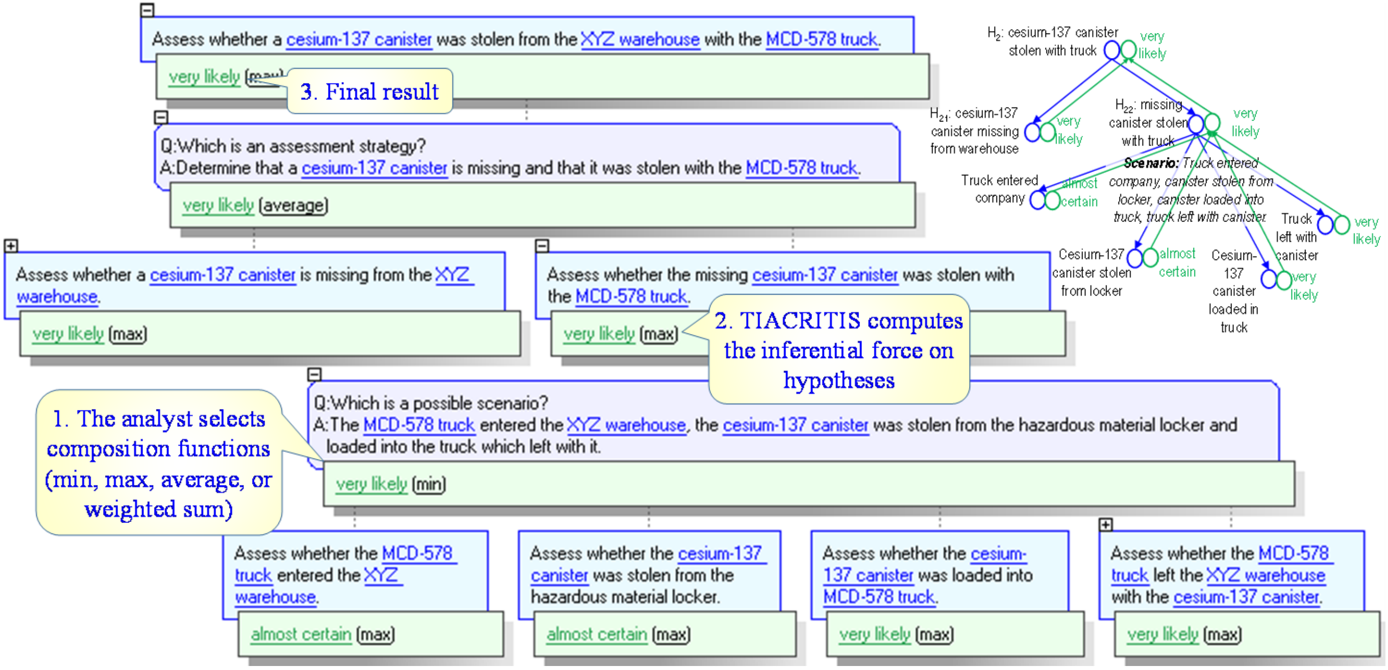
After all the elementary hypotheses have been assessed, either based on evidence or by making assumptions, the user has to select the solution composition functions (e.g., min, max, average, or weighted sum) to be used by TIACRITIS when assessing the likelihoods of the intermediary hypotheses and of the top level hypothesis, as shown in Fig. 9.

TIACRITIS not only supports the analyst in hypotheses analysis, but it also continuously learns to facilitate the analysis of new hypotheses. Consider, for examples, the new hypothesis analyses problem from the top of Fig. 10. TIACRITIS suggests a reduction based on a pattern learned from the analysis in Fig. 5. It also suggests the question for another assessment strategy to be defined by the analyst. Of course, the more TIACRITIS learns, the more useful its suggestions.

V. Final Remarks

TIACRITIS is an operational web-based system, and is available for education and analysis (see Fig. 11). It includes modules from the Disciple Learning Agent Shell, as well as modules that implement the current version of the computational theory of intelligence analysis. Its use is supported by three textbooks and numerous case studies:

- "Introduction to Intelligence Analysis: A Hands-on



Assess whether a cesium-137 canister was stolen from the XYZ warehouse with the MCD-578 truck.

very likely (max)

3. Final result

Q:Which is an assessment strategy?
A:Determine that a cesium-137 canister is missing and that it was stolen with the MCD-578 truck.

very likely (average)

Assess whether a cesium-137 canister is missing from the XYZ warehouse.

very likely (max)

Assess whether the missing cesium-137 canister was stolen with the MCD-578 truck.

very likely (max)

2. TIACRITIS computes the inferential force on hypotheses

1. The analyst selects composition functions (min, max, average, or weighted sum)

Q:Which is a possible scenario?
A:The MCD-578 truck entered the XYZ warehouse, the cesium-137 canister was stolen from the hazardous material locker and loaded into the truck which left with it.

very likely (min)

Assess whether the MCD-578 truck entered the XYZ warehouse.

almost certain (max)

Assess whether the cesium-137 canister was stolen from the hazardous material locker.

almost certain (max)

Assess whether the cesium-137 canister was loaded into MCD-578 truck.

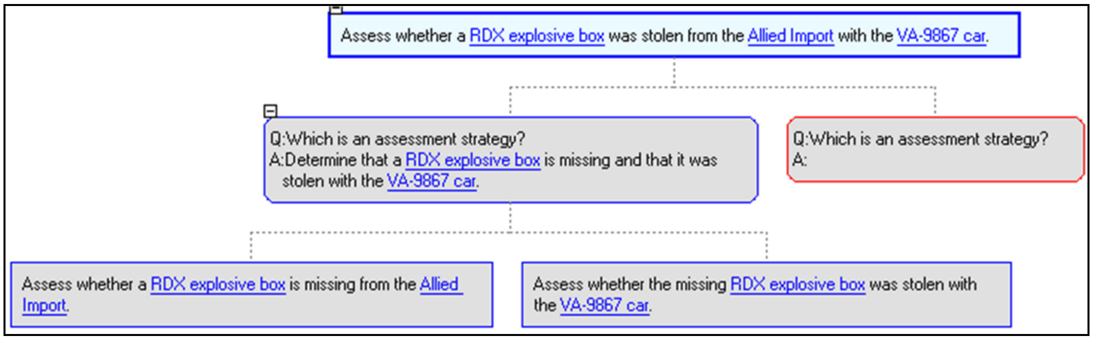
very likely (max)

Assess whether the MCD-578 truck left the XYZ warehouse with the cesium-137 canister.

very likely (max)

H_2: cesium-137 canister stolen with truck
very likely
H_{21}: cesium-137 canister missing from warehouse
very likely
H_{22}: missing canister stolen with truck
very likely
Scenario: Truck entered company, canister stolen from locker, canister loaded into truck, truck left with canister.
Truck entered company
almost certain
Cesium-137 canister stolen from locker
almost certain
Cesium-137 canister loaded in truck
very likely
Truck left with canister
very likely

Figure 9. The top part of the hypothesis analysis tree showing the solution composition functions and the likelihoods of the hypotheses.



Assess whether a RDX explosive box was stolen from the Allied Import with the VA-9867 car.

Q:Which is an assessment strategy?
A:Determine that a RDX explosive box is missing and that it was stolen with the VA-9867 car.

Q:Which is an assessment strategy?
A:

Assess whether a RDX explosive box is missing from the Allied Import.

Assess whether the missing RDX explosive box was stolen with the VA-9867 car.

Figure 10. Reductions suggested by TIACRITIS based on learned analysis patterns.

Approach with TIACRITIS” teaches basic knowledge about the properties, uses, and marshaling of evidence to show students how to collect evidence and test hypotheses by assessing the relevance, the believability, and the inferential force of evidence [2].

- “A Practicum in Evidence Marshaling and Argument Construction with TIACRITIS” teaches advanced strategies for organizing and combining analyst’s thoughts and evidence to construct complex arguments from masses of evidence (in preparation).
- “Modeling Violent Extremists with TIACRITIS” teaches an evidence-based methodology for investigating, comprehending, and anticipating the behavior of violent extremists in the war on terror [13].

One main direction of follow-on work is further development of the computational theory and its implementation in TIACRITIS. This includes the development of computational models for evidence marshaling guided by magnets which are powerful heuristics supporting the analysts in hypotheses generation from masses of evidence. Future research also includes the development of more powerful methods for the learning and reuse of analytic expertise, for hypotheses generation through mixed-initiative abduction, for collaborative analysis, for automatic report generation, and for decision-making under uncertainty which integrates the computational theory. Although the focus of the current work was on mixed-initiative analysis involving analysts, TIACRITIS and the theory it is built on can be extended to persistent surveillance and interpretation of dynamic environments by autonomous agents.

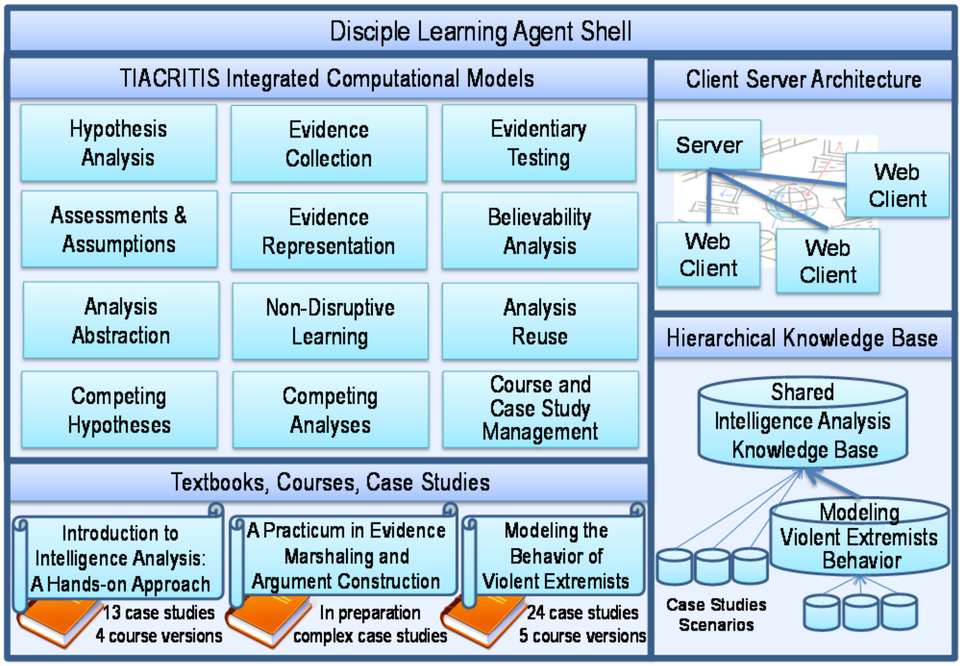

Disciple Learning Agent Shell
TIACRITIS Integrated Computational Models
Hypothesis Analysis
Evidence Collection
Evidentiary Testing
Assessments & Assumptions
Evidence Representation
Believability Analysis
Analysis Abstraction
Non-Disruptive Learning
Analysis Reuse
Competing Hypotheses
Competing Analyses
Course and Case Study Management
Client Server Architecture
Server
Web Client
Web Client
Web Client
Hierarchical Knowledge Base
Shared Intelligence Analysis Knowledge Base
Modeling Violent Extremists Behavior
Case Studies Scenarios
Textbooks, Courses, Case Studies
Introduction to Intelligence Analysis: A Hands-on Approach
13 case studies
4 course versions
A Practicum in Evidence Marshaling and Argument Construction
In preparation
complex case studies
Modeling the Behavior of Violent Extremists
24 case studies
5 course versions

Figure 11. TIACRITIS cognitive assistant and textbooks.

REFERENCES

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

[2] Tecuci G., Schum D.A., Boicu M., Marcu D. (2010). *Introduction to Intelligence Analysis: A Hands-on Approach with TIACRITIS,* 220 pages, George Mason University, new edition 2011.

[3] Tecuci G., Boicu M., Cox M.T. (2007). Seven Aspects of Mixed-initiative Reasoning: An Introduction to the Special Issue on Mixed-initiative Assistants. *AI Magazine* 28(2), 11–18.

[4] Schum D.A. (2009). Science of Evidence: Contributions from Law and Probability. *Law Probab Risk* 8, 197–231.

[5] Tecuci G., Boicu M., Marcu D., Schum D., Hamilton B. (2010). TIACRITIS System and Textbook: Learning Intelligence Analysis through Practice, in *Proc of the 5th Int. Conf. on Semantic Technologies for Intelligence, Defense, and Security – STIDS,* 108-115, Fairfax, VA.

[6] Nilsson N.J. (1971). *Problem Solving Methods in Artificial Intelligence*. NY: McGraw-Hill.

[7] Tecuci G. (1988). *DISCIPLE: A Theory, Methodology and System for Learning Expert Knowledge*. Thèse de Docteur en Science, University of Paris-South, France.

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

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

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

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

[12] Weiss C. (2008). Communicating Uncertainty in Intelligence and Other Professions, *International Journal of Intelligence and CounterIntelligence*, 21(1), 57–85.

[13] Boicu M., Tecuci G., Marcu D., Schum D.A., Coughlin S. (2011). *Modeling Violent Extremists with TIACRITIS,* 203 pages, George Mason University, 2011.

PR-OWL 2 Case Study: A Maritime Domain Probabilistic Ontology

Kathryn Blackmond Laskey, Richard Haberlin,
Paulo Costa
Volgenau School of Engineering
George Mason University
Fairfax, VA USA
[klaskey, rhaberli, pcosta]@gmu.edu

Rommel Novaes Carvalho
Brazilian Office of the Comptroller General
Brasília, Brazil
rommel.carvalho@gmail.com

***Abstract*—Probabilistic ontologies incorporate uncertain and incomplete information into domain ontologies, allowing uncertainty in attributes of and relationships among domain entities to be represented in a consistent and coherent manner. The probabilistic ontology language PR-OWL provides OWL constructs for representing multi-entity Bayesian network (MEBN) theories. Although compatibility with OWL was a major design goal of PR-OWL, the initial version fell short in several important respects. These shortcomings are addressed by the latest version, PR-OWL 2. This paper provides an overview of the new features of PR-OWL 2 and presents a case study of a probabilistic ontology in the maritime domain. The case study describes the process of constructing a PR-OWL 2 ontology using an existing OWL ontology as a starting point.**

Keywords- Probabilistic ontology, Multi-Entity Bayesian networks, PR-OWL, OWL, Maritime domain ontology, Uncertainty Modeling Process for Semantic Technologies

I. INTRODUCTION

The emphasis on net-centric operations and the shift to asymmetric warfare have created new challenges for automated information integration. To meet these challenges, developers are recognizing the need to combine explicit representation of domain semantics with the ability to represent and reason with uncertainty. Probabilistic ontologies allow the representation of uncertainty about attributes of and relationships among domain entities. Probabilistic OWL (PR-OWL) [1] is an OWL upper ontology for representing probabilistic ontologies. Compatibility with OWL was a major design goal for PR-OWL. However, the initial release of PR-OWL falls short of complete compatibility in several important respects. First, there is no mapping in PR-OWL to properties of OWL. Second, although PR-OWL has the concept of meta-entities, which allows the definition of complex types, it lacks compatibility with existing types already present in OWL. These problems have been noted in the literature [2]:

> PR-OWL does not provide a proper integration of the formalism of MEBN and the logical basis of OWL on the meta level. More specifically, as the connection between a statement in PR-OWL and a statement in OWL is not formalized, it is unclear how to perform the integration of ontologies that contain statements of both formalisms.

Carvalho [3] proposed a new syntax and semantics, defined as PR-OWL 2, which improves compatibility between PR-OWL and OWL in two important respects. First, PR-OWL 2 follows the approach suggested by Poole et al. to formalizing the association between random variables from probabilistic theories with the individuals, classes and properties from ontological languages such as OWL. Second, PR-OWL 2 allows values of random variables to range over OWL datatypes.

This paper presents an overview of PR-OWL 2, describes the key features that improve compatibility with OWL, discusses an open-source tool for building PR-OWL 2 probabilistic ontologies, and describes a use case of a PR-OWL 2 ontology for maritime domain awareness.

II. A PROBABILISTIC ONTOLOGY IN PR-OWL

A. PR-OWL 1: An Upper Ontology for MEBN Theories

PR-OWL provides constructs to define probabilistic ontologies in the OWL ontology language. The initial version, PR-OWL 1, is an OWL upper ontology for representing MEBN theories [4]. MEBN is a first-order probabilistic language (FOPL) [5] that allows probabilities to be assigned in a consistent way to logical statements. MEBN represents the world as entities that have attributes and are related to other entities. Knowledge about the attributes of entities and their relationships to each other is represented as a collection of MEBN fragments (MFrags) organized into MEBN Theories (MTheories). An MFrag represents a conditional probability distribution for instances of its resident random variables given their parents in the fragment graph and the context nodes. An MTheory is a set of MFrags that collectively satisfies consistency constraints ensuring the existence of a unique joint probability distribution over instances of the random variables represented in each of the MFrags within the set. A PR-OWL ontology encodes domain knowledge as a set of MFrags. A PR-OWL reasoner uses the probability information encoded in the MFrags to compute responses to probabilistic queries.

This research was partially supported by the Office of Naval Research (ONR), under Contract N00173-09-C-4008. Rommel Carvalho would like to thank the Brazilian Office of the Comptroller General (CGU) for their active support since 2008 and for providing the human resources necessary to conduct this research.

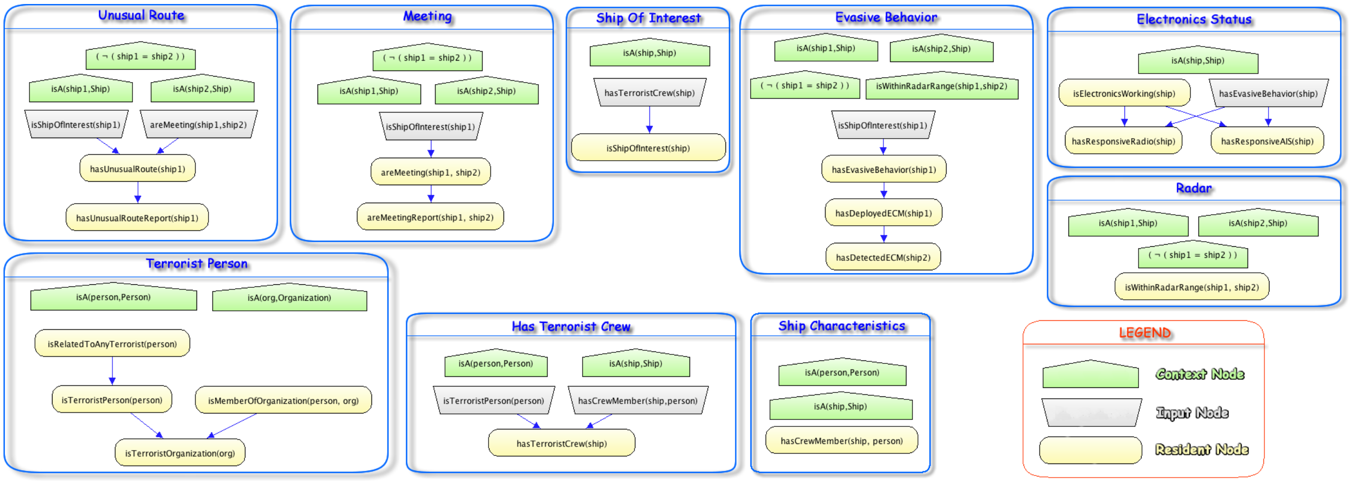


Figure 1. Probabilistic Ontology for Identifying Ship-of-Interest

B. A PR-OWL Ontology for the Maritime Domain

As an example of a PR-OWL ontology, Figure 1 shows a simple probabilistic ontology developed as part of the PROGNOS (Probabilistic OntoloGies for Net-centric Operation Systems) project [6]. The ontology is designed for the problem of identifying whether a vessel is a ship of interest. The model is designed to answer the following queries using the following evidence:

***Overall Goal**: Identify whether a ship is a ship of interest, i.e. if the ship seems to be suspicious in any way.*

1. **Query**: Does the ship have a terrorist crewmember?
   a. **Evidence**: Verify whether a crewmember is related to any terrorist;
   b. **Evidence**: Verify whether a crewmember is associated with any terrorist organization.
2. **Query**: Is the ship using an unusual route?
   a. **Evidence**: Verify whether there is a direct report that the ship is using an unusual route;
   b. **Evidence**: Verify whether there is a report that the ship is meeting some other ship for no apparent reason.
3. **Query**: Does the ship seem to exhibit evasive behavior?
   a. **Evidence**: Verify whether an electronic countermeasure (ECM) was identified by a navy ship;
   b. **Evidence**: Verify whether the ship has a responsive radar and automatic identification system (AIS).

Each of the nine MFrags of Figure 1 addresses a modular component of the knowledge needed to address the above queries. Specifically, probabilistic knowledge about hypotheses related to the identification of a terrorist crewmember is represented in the *HasTerroristCrew*, *TerroristPerson*, and *ShipCharacteristics* MFrags. Knowledge about unusual routes is represented in the *UnusualRoute* and *Meeting* MFrags. Finally, knowledge about hypotheses related to evasive behavior is represented in the *EvasiveBehavior*, *Eletronics-Status*, and *Radar* MFrags.

A detailed explanation of this model can be found in [6]. The model was expanded and extended iteratively as described in [7] to address additional queries and evidence.

C. An Open Source Tool for Probabilistic Ontologies

The MFrags shown in Figure 1 are screenshots from the UnBBayes-MEBN [8], an open source, plug-in-based Java application for building and reasoning with probabilistic ontologies based on the PR-OWL/MEBN framework.[1] It features a graphical user interface (GUI), an application programming interface (API) for saving and loading PR-OWL ontologies, reasoning algorithms for processing queries, and plugin support for extensions.

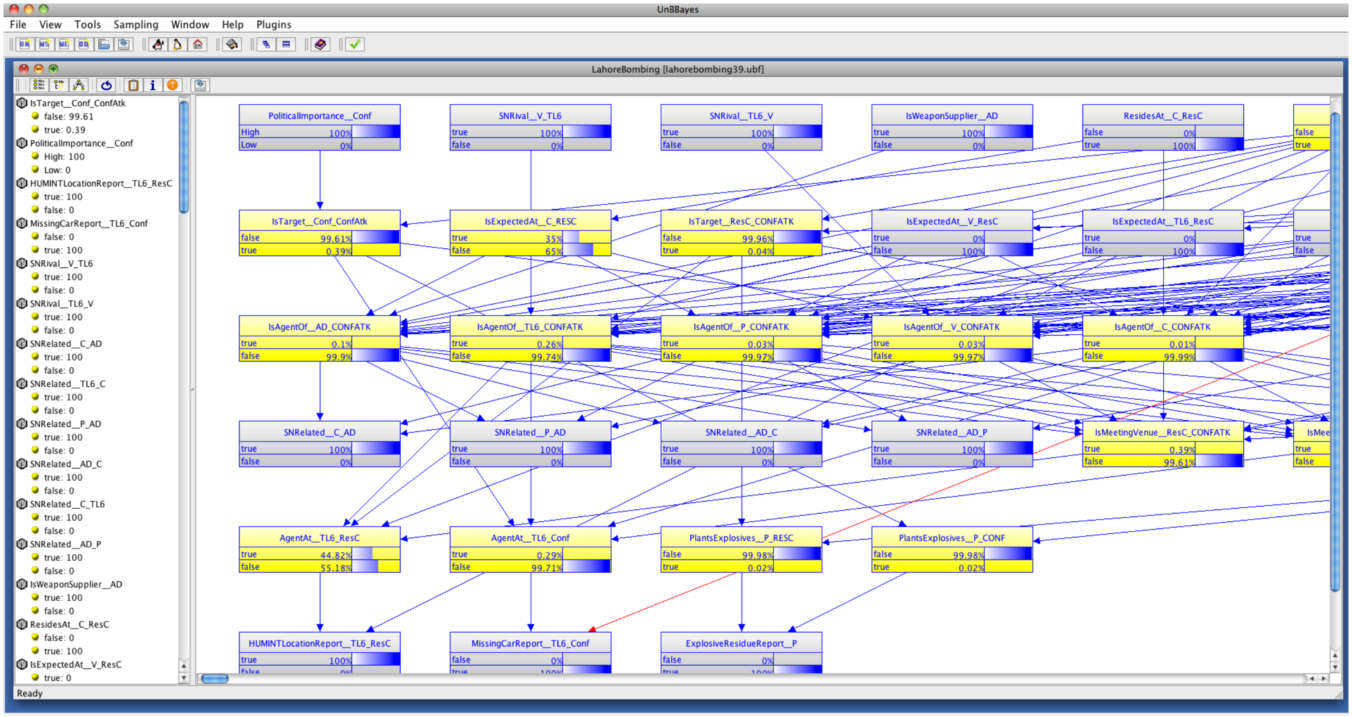
D. Queries

Queries are processed in UnBBayes-MEBN using an implementation of the situation-specific Bayesian network (SSBN) construction algorithm described in [4]. Figure 2 shows an SSBN built using the implemented algorithm. We applied an exact inference algorithm on small-scale problems to test the model and identify logical inconsistencies, differences in query results from those expected by subject-matter experts, and other flaws in the model. For larger scale problems, approximate inference algorithms are employed to mitigate scalability issues. We also implemented hypothesis management methods [9] to control the complexity of the constructed networks while maintaining acceptable accuracy in results.

III. PR-OWL 2: Improving Compatibility with OWL

Ideally, it should be possible to use PR-OWL to reason probabilistically about uncertain aspects of an ontology based on the information already available. That is, we would like to

[1] UnBBayes is available from http://unbbayes.sourceforge.net/


UnBBayes
File View Tools Sampling Window Help Plugins
LahoreBombing [lahorebombing39.ubf]
PoliticalImportance__Conf High 100% Low 0%
SNRival__V_TL6 true 100% false 0%
SNRival__TL6_V true 100% false 0%
IsWeaponSupplier__AD true 100% false 0%
ResidesAt__C_ResC false 0% true 100%
IsTarget__Conf_ConfAtk false 99.61% true 0.39%
IsExpectedAt__C_RESC true 35% false 65%
IsTarget__ResC_CONFATK false 99.96% true 0.04%
IsExpectedAt__V_ResC true 0% false 100%
IsExpectedAt__TL6_ResC true 0% false 100%
IsAgentOf__AD_CONFATK true 0.1% false 99.9%
IsAgentOf__TL6_CONFATK true 0.26% false 99.74%
IsAgentOf__P_CONFATK true 0.03% false 99.97%
IsAgentOf__V_CONFATK true 0.03% false 99.97%
IsAgentOf__C_CONFATK true 0.01% false 99.99%
SNRelated__C_AD true 100% false 0%
SNRelated__P_AD true 100% false 0%
SNRelated__AD_C true 100% false 0%
SNRelated__AD_P true 100% false 0%
IsMeetingVenue__ResC_CONFATK true 0.39% false 99.61%
IsMee true false
AgentAt__TL6_ResC true 44.82% false 55.18%
AgentAt__TL6_Conf true 0.29% false 99.71%
PlantsExplosives__P_RESC false 99.98% true 0.02%
PlantsExplosives__P_CONF false 99.98% true 0.02%
HUMINTLocationReport__TL6_ResC true 100% false 0%
MissingCarReport__TL6_Conf false 0% true 100%
ExplosiveResidueReport__P false 0% true 100%
Ready

IsTarget__Conf_ConfAtk
false: 99.61
true: 0.39
PoliticalImportance__Conf
High: 100
Low: 0
HUMINTLocationReport__TL6_ResC
true: 100
false: 0
MissingCarReport__TL6_Conf
false: 0
true: 100
SNRival__V_TL6
true: 100
false: 0
SNRival__TL6_V
true: 100
false: 0
SNRelated__C_AD
true: 100
false: 0
SNRelated__TL6_C
true: 100
false: 0
SNRelated__P_AD
true: 100
false: 0
SNRelated__AD_C
true: 100
false: 0
SNRelated__C_TL6
true: 100
false: 0
SNRelated__AD_P
true: 100
false: 0
IsWeaponSupplier__AD
true: 100
false: 0
ResidesAt__C_ResC
false: 0
true: 100
IsExpectedAt__V_ResC
true: 0

Figure 2. Situation-Specific Bayesian Network for Identifying Ship-of-Interest

be able to begin with an OWL ontology containing information about a domain, use PR-OWL to define uncertainty about attributes of and relationships among the entities, and apply a probabilistic reasoner to reason with available evidence. For example, we might begin with an OWL ontology containing classes for ships, routes, persons, and other entities mentioned in the MFrags of Figure1. We would then wish to use PR-OWL to define the probability distributions represented in the MFrags.

The difficulty with this idea is that PR-OWL 1 has no mapping between the random variables used in PR-OWL and the properties used in OWL. For example, suppose we have defined an OWL class Ship with property `isShipOfInterest`, intended to represent whether a ship is a ship-of-interest. We might want to use the PR-OWL random variable `isShipOfInterest(ship)` to define the uncertainty associated with this property. We might use the *ShipOfInterest* MFrag of Figure 1 to specify its probability distribution. However, despite the syntactic similarity between the property name and the random variable name, PR-OWL 1 has no way to specify formally that the random variable `isShipOfInterest(ship)` defines the uncertainty of the OWL property `isShipOfInterest`. Thus, even if we had information about whether a particular ship, say `Ship379`, is a ship-of-interest, we would not be able to instantiate the random variable `isShipOfInterest(ship)` for `Ship379`.

Poole et al. [10] point out the need to relate the random variables from probabilistic theories to the individuals, properties and classes of ontological languages like OWL. Poole et al. state, "We can reconcile these views by having properties of individuals correspond to random variables." This is the approach taken in PR-OWL 2.

The key to building the bridge that connects the deterministic ontology defined in OWL and its probabilistic extension defined in PR-OWL is to understand how to translate one to the other. On the one hand, given a concept defined in OWL, how should its uncertainty be defined in PR-OWL in a way that maintains its semantics defined in OWL? On the other hand, given a random variable defined in PR-OWL, how should it be represented in OWL in a way that respects its uncertainty already defined in PR-OWL?

PR-OWL 2 formalizes the relationship between OWL properties and PR-OWL random variables using the relation `definesUncertaintyOf` [3]. In our previous example, we would use the relation `definesUncertaintyOf` [3] to relate the OWL property `isShipOfInterest` to the PR-OWL 2 random variable `isShipOfInterest(ship)`. An additional complexity arises because MEBN can represent *n*-ary functions and predicates, whereas OWL has only binary properties. We must ensure that not only is the random variable linked to its associated OWL property by `definesUncertaintyOf`, but also its arguments are linked to their respective OWL properties by either `isSubjectIn` or `isObjectIn`, depending on whether they refer to the domain or range of the OWL property, respectively. This feature is especially important when dealing with n-ary random variables, where each argument of the random variable will be associated with a different OWL property.

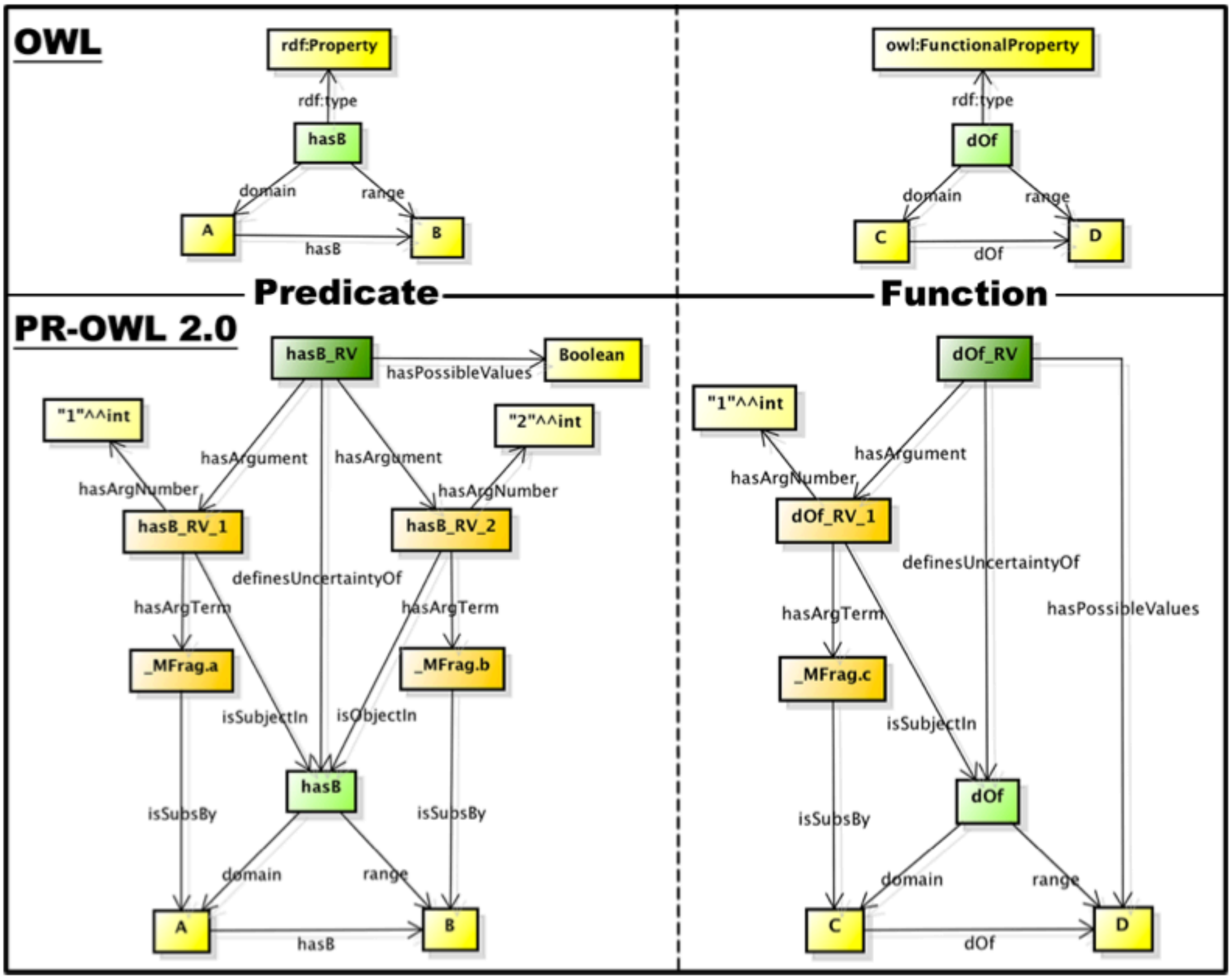


Figure 3. Mapping of PR-OWL Random Variables and OWL Properties

Figure 3 shows a schematic for the mapping between OWL properties and PR-OWL random variables. A full discussion of the formal mapping between OWL properties and PR-OWL random variables can be found in [3]. The mapping provides the basis for a formal definition of consistency between a PR-OWL probabilistic ontology and an OWL ontology, in which rules in the OWL ontology correspond to probability one assertions in the PR-OWL ontology. A formal notion of consistency can lead to development of consistency checking algorithms.

Another major difference between PR-OWL 1 and PR-OWL 2 is that the separate definition of entity in PR-OWL is replaced by OWL's built-in notion of classes and data types. That is, a PR-OWL entity is now identified with either a class or a data type in OWL. Moreover, since OWL supports multiple inheritance, so does PR-OWL 2. Thus, all the control over the type definition and type hierarchy in PR-OWL is delegated to OWL.

In PR-OWL 2, therefore, the possible values or outcomes of a random variable are instances of classes and data types. When specifying that a random variable will have individuals of a class as its possible outcomes, it is reasonable to assume that all known individuals of that class form a set of collectively exhaustive outcomes. However, the assumptions about individuals in OWL are not enough to guarantee these individuals are mutually exclusive. More specifically, although OWL provides a way to express unique names, it also allows two different names to point to the same object in the real world. To address this issue, PR-OWL 2 follows the MEBN and PR-OWL 1 convention, and assumes that every individual has a unique ID associated to it.

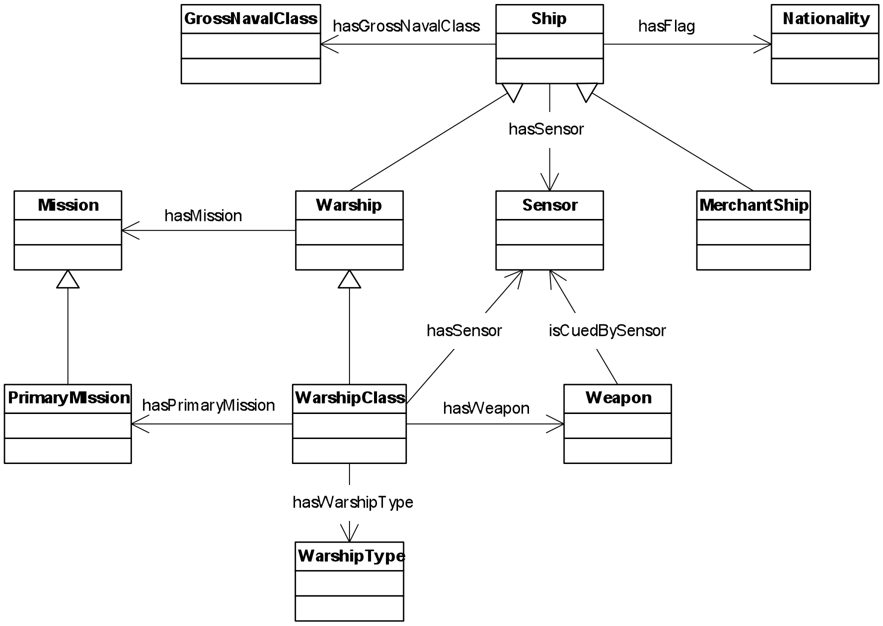


Figure 5. Entity-Relationship Diagram for Maritime Ship Ontology

We note that there are certain aspects of the full PR-OWL semantics that are not fully captured in OWL-DL, and therefore cannot be handled by OWL-DL reasoners, but are expected to be respected by PR-OWL reasoners. In particular, to specify the restriction that a random variable defines the uncertainty of a property would require OWL Full. For this reason, the restriction is not explicitly represented in PR-OWL, but it is expected to be enforced by a PR-OWL probabilistic reasoner. This enables consistency checking of the deterministic part of a PR-OWL ontology using a DL reasoner.

IV. PR-OWL 2 CASE STUDY

The following case study demonstrates the application of probability to an existing ontology to represent uncertainty in knowledge about instance attributes. In this case, an existing ontology of Western European warships identifies the major characteristics of each combatant class through the attributes of size, sensors, weapons, missions, and nationality. Figure 5 shows an entity-relationship diagram for the ontology. The decision maker is trying to determine the warship class of a contact about which he has limited information. By adding probability to the existing ontology, we can identify the most likely class of ship he is encountering when provided only partial or uncertain information. The model is designed to answer the following query using the following evidence:

***Overall Goal**: Given uncertain or absent attribute information about a specific ship, what is the most likely European warship class that satisfies these attributes?*

1. **Query**: What is the type of warship?
   a. **Evidence**: Identify the size of the ship;
   b. **Evidence**: Confirm the ship is a warship;
   c. **Evidence**: Identify the primary mission of the ship based on its weapons and sensors.
2. **Query**: What nation has flagged the ship?
   a. **Evidence**: Identify the nation under which the ship is registered.

The entity-relationship diagram of Figure 5 presents a simplified design of the Military Ship Ontology illustrating the primary attributes used to answer these queries. The decision maker desires to know the class of warship that he faces. A class of ships has a consistent hull design and a standardized suite of weapons and sensors. These weapons and sensors work in concert to provide synergy in executing the primary mission of each type of ship. By combining a ship type with the nation that operates it, a logical prediction of warship class may be obtained.

International law of the sea requires that each merchant ship is registered and sails under a single nation for the purpose of regulation, certification, and pollution control. That process is known as flagging, and an individual ship is flagged by a nation. It is not required that a ship is flagged under the same nation as its owner; a “flag of convenience” allows a ship to be operated under an alternate nation to reduce operating costs and regulations. However, warships are always flagged under the nation of ownership.

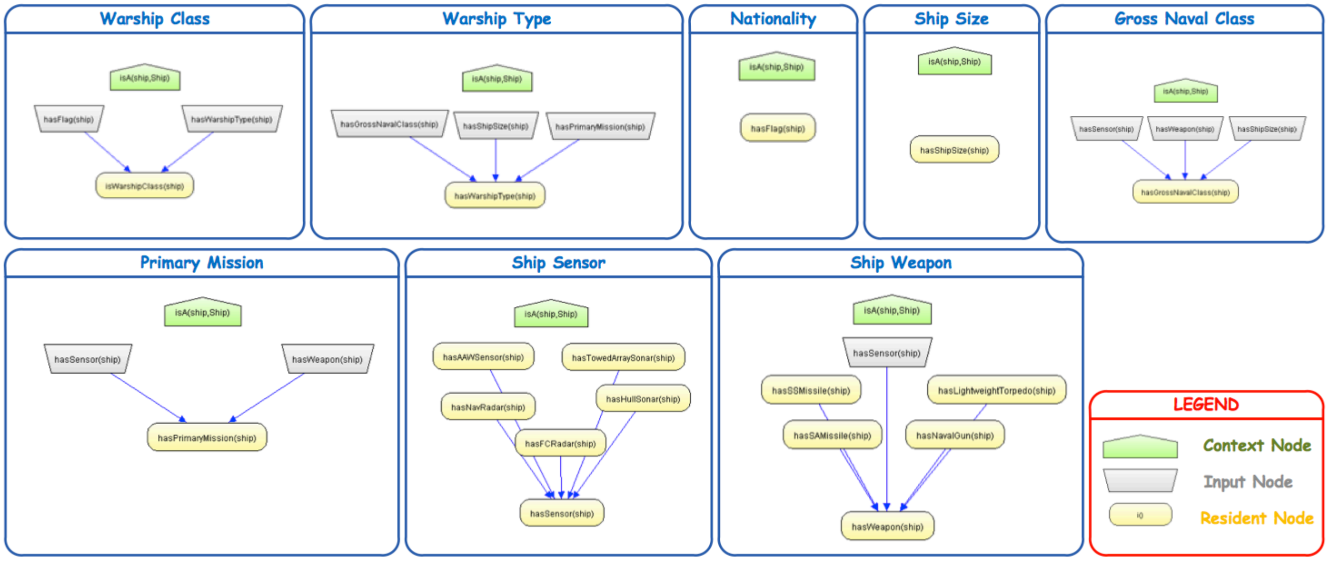


Figure 6. Military Ship Probabilistic Ontology

The Gross Naval Class is a naval schema that delineates warships from merchant ships, and is mutually exclusive. Through identification of weapon and sensor attributes, as well as overall ship size, a Gross Naval Class estimate may be made for the unknown ship. While it can be assumed that all ships have a radar sensor, only military ships have sensors associated with weapons systems. The presence of a weapon system, or a weapon-associated sensor, provides reasonable evidence that a ship is a warship.

Warships are of different types based on their primary mission. Most ships have multiple mission capabilities, but for this ontology we assume the following primary mission areas by ship type:

Anti-Air Warfare (AAW):
- Aircraft Carrier (CV, CVN)
- Cruiser (CG)
- Guided Missile Destroyer (DDG)
- Guided Missile Frigate (FFG)

Anti-Surface Warfare (ASuW):
- Destroyer (DD)

Anti-Submarine Warfare (ASW):
- Frigate (FF)

By observing the combination of weapons and sensors, it is possible to infer the most likely mission area. This, combined with an estimate of ship size, provides an indication of the type of warship.

At this point an MTheory is created to determine `hasWarshipClass(ship)` in the *WarshipClass* MFrag for some unknown ship. The eight MFrags associated with this determination are shown in Figure 6. Inputs to `hasWarshipClass` RV are the RVs from the *WarshipType* and *Nationality* MFrags, representing the concepts introduced above with the RVs `hasWarshipType(ship)` and `hasFlag(ship)`. The *WarshipType* MFrag may be further decomposed into the *ShipSize*, *GrossNavalClass*, and *PrimaryMission* MFrags. The *GrossNavalClass* MFrag is influenced by both the *ShipSize* and *ShipSensor* MFrags through the `hasShipSize(ship)` and `hasSensor(ship)` RVs, while the *PrimaryMission* MFrag is influenced by the *ShipSensor* and *ShipWeapon* MFrags with `hasSensor(ship)` and `hasWeapon(ship)` RVs. With the MTheory complete as shown in Figure 6, the Local Probability Distribution (LPD) must be populated.

Prior probabilities for the `hasFlag` RV were obtained from an estimate of merchant ship registrations available through open source information. Similarly, `hasShipSize` represents a finite and exhaustible set of ship lengths (`LengthLess150m, Length150to-100m, LengthGreater200m`) into which each `ship` is categorized. Prior probability estimates were again obtained via open source literature. Priors for `hasSensor` and `hasWeapon` were obtained through subject-matter-expert review of open source literature and represent the proportion of warships with each of the types of sensors. LPDs for the *GrossNavalClass* and *PrimaryMission* MFrags require conditional statements about relationships from the input nodes shown in Figure 6. A detailed description of these relationships is described in a forthcoming paper.

Queries to the Military Ship Probabilistic Ontology are processed in UnBBayes-MEBN using an implementation of the situation-specific Bayesian network (SSBN) construction algorithm. Instances of unknown ships and representative evidence are entered via the OWL ontology through the UnbBayes GUI to reflect partial or uncertain information

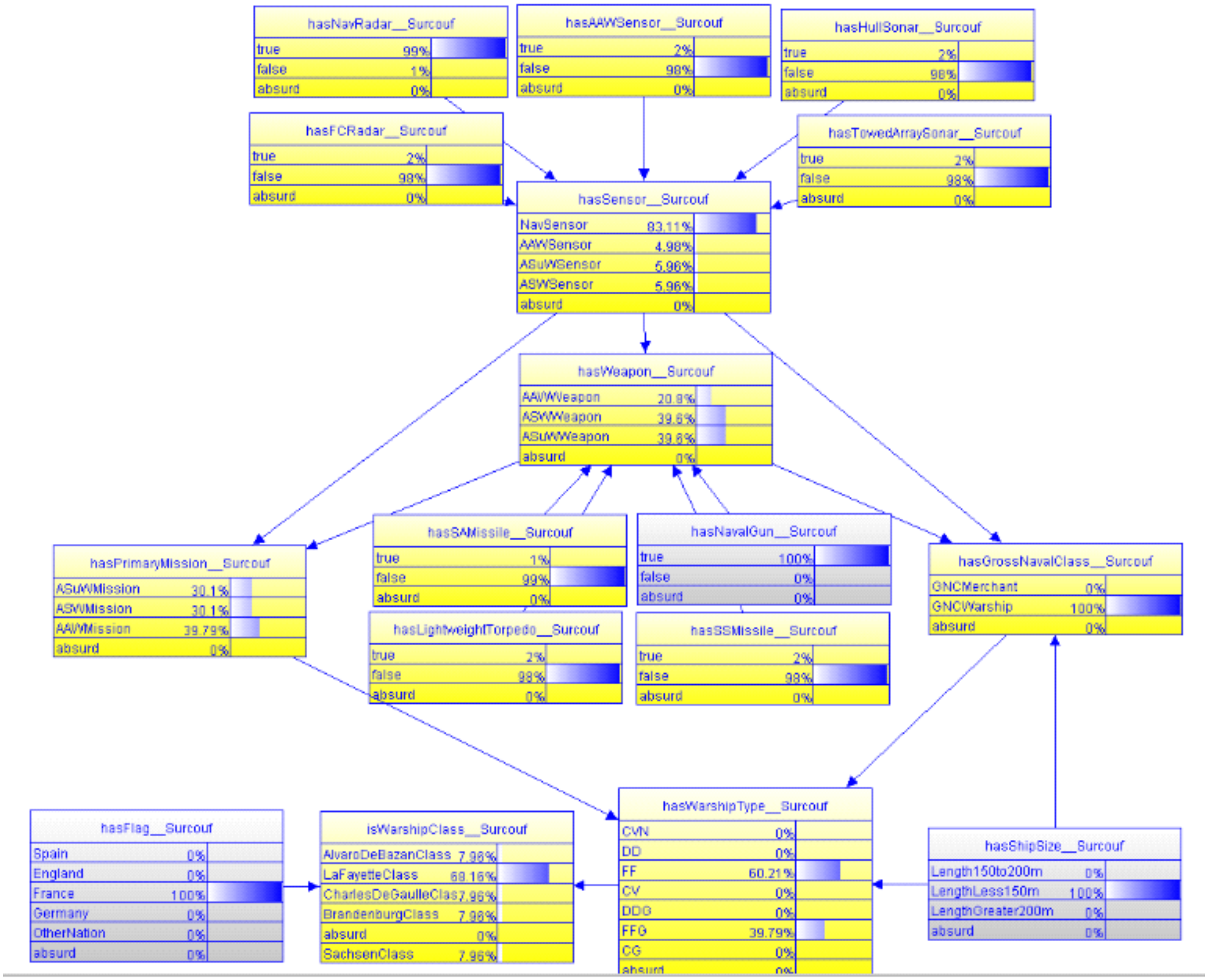


Figure 7 Situation-Specific Bayesian Network Military Ship Classification

about ship attributes. These are checked against known characteristics provided by subject-matter experts.

For example, suppose the following evidence is obtained about a ship of interest:

- `UID: Surcouf`
- `hasNavalGun(Surcouf): True`
- `hasFlag(Surcouf): France`
- `hasShipSize(Surcouf): <150m`

Executing a query of the `isWarshipClass` node produces the SSBN found in Figure 7. In this case, there is a 68% chance that *Surcouf* is a member of the French LaFayette Class of frigates, which is the correct classification.

As discussed in Section III, our goal is to begin with an OWL ontology containing information about a domain, use PR-OWL to define uncertainty about attributes of and relationships among the entities, and apply a probabilistic reasoner to reason with available evidence. Using the formalized construct introduced in PROWL-2, we map each of the RVs in the MFrags of the probabilistic ontology to the existing OWL property in the original ontology. This is accomplished through the probabilistic ontology building sequence executed on the UnbBayes software. For example, the `WarshipType` class in OWL has an object property of `hasPrimaryMission`. This object property is mapped to the `hasPrimaryMission(ship)` RV of the *PrimaryMission* MFrag. Mappings produced for each RV and its associated property in OWL allow us to use PR-OWL to reason probabilistically about uncertain aspects of an existing ontology based on the information already available.

V. CONCLUSION

Combining uncertainty reasoning with semantic technology is necessary for robust, interoperable, net-centric fusion and decision support systems. The probabilistic ontology language PR-OWL provides a way to represent and reason with probabilistic ontologies. PR-OWL 2 improves compatibility with OWL in several important respects. Through a case study, this paper describes the construction of a probabilistic ontology obtained by enhancing an existing OWL ontology with probability information.

References

[1] P. C. G. Costa, *Bayesian semantics for the semantic web*, PhD dissertation, Fairfax, VA, George Mason University (Jul. 2005).

[2] L. Predoiu, H. Stuckenschmidt, Probabilistic extensions of semantic web languages - a survey, in: *The Semantic Web for Knowledge and Data Management: Technologies and Practices*, Idea Group Inc, 2008.

[3] R. N. Carvalho, Probabilistic ontology: Representation and modeling methodology, PhD dissertation, Fairfax, VA, George Mason University (Jun. 2011).

[4] K. B. Laskey, MEBN: a language for First-Order bayesian knowledge bases, *Artificial Intelligence* 172 (2-3) (2008) 140–178.

[5] B. Milch, S. Russell, First-Order probabilistic languages: Into the unknown, in: Inductive Logic Programming, Vol. 4455 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2007, pp. 10–24.

[6] R. N. Carvalho, P. C. G. Costa, K. B. Laskey, and K. Chang, “PROGNOS: predictive situational awareness with probabilistic ontologies,” in *Proceedings of the 13th International Conference on Information Fusion*, Edinburgh, UK, Jul. 2010.

[7] Carvalho, R.N., Haberlin, R., Costa, P., Laskey, K.B. and Chang, K.C., Modeling a Probabilistic Ontology for Maritime Domain Awareness. *Proceedings of the Fourteenth International Conference on Information Fusion*, July 2011.

[8] Matsumoto, S., Carvalho, R., Costa, P., Laskey, K.B., Santos, L.L. and Ladeira, M. There’s No More Need to be a Night OWL: on the PR-OWL for a MEBN Tool Before Nightfall. in *Introduction to the Semantic Web: Concepts, Technologies and Applications*, G. Fung, Ed. iConcept Press, 2011.

[9] R. Haberlin, P. C. G. da Costa, K. B. Laskey, Hypothesis management in support of inferential reasoning, in: Proceedings of the Fifteenth Inter- national Command and Control Research and Technology Symposium, Santa Monica, CA, USA, 2010.

[10] D. Poole, C. Smyth, R. Sharma, Semantic science: Ontologies, data and probabilistic theories, in: Uncertainty Reasoning for the Semantic Web I, Vol. 5327 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2008, pp. 26–40.

Position Papers

An Ontology-based Adaptive Reporting Tool

Christian Mårtenson, Andreas Horndahl
Swedish Defense Research Agency (FOI)
Stockholm, Sweden
firstname.lastname(at)foi .se

Ziaul Kabir
The Royal Institute of Technology (KTH)
Stockholm, Sweden
mzkabir(at)kth.se

Abstract— **Intelligence gathering by human observers is important for acquiring indirect and non-physical information. The drawback is that it is often delivered as free text which is not well-suited for further exploitation through automatic processing. In this paper we present a concept for structured human reporting based on an ontology-driven adaptive user-interface. The concept lays the foundation for the implementation of a possibly hand-held in-field reporting system, which can adapt to the context of the reporting situation as well as to possible information needs of other agents in the intelligence system.**

Keywords-semantic technologies; ontologies; adaptive user interfaces; context aware interaction

I. Introduction

In spite of constant technological advances, the nature of today's conflicts has increased the importance of intelligence gathering by human observers. Automatic sensing systems do a good job detecting and monitoring physical features like vehicle or human movements, but for acquiring indirect information and information referring to the cognitive domain humans are still the main asset. This kind of information is often referred to as soft data. The advantage of soft data is its high informational value; the drawback is that it is often delivered as free text, which though human friendly is less suitable for exploitation through automatic processing. Hence an important issue in managing soft data is the transformation of unstructured free text into structured content adhering to a formalized information model. Techniques for automatic structuring of text include linguistic and statistical approaches for entity and relation extraction. Such techniques are computational intense, often require a lot of training data and are never completely accurate. In a human reporting system these are limiting factors and alternative approaches are of interest.

One might argue that speaking or writing in your native tongue is the most intuitive method for delivering a human message, and that issues regarding human reporting will be solved when language processing has been cultivated to perfection or near perfection. However, the opposite approach, forcing the human reporter to directly input structured information can have other benefits:

- The language is more precise, which can prevent the user from making unintentional fuzzy statements
- The format is more compact, implying a potential for faster input
- The underlying information model is based on a shared understanding, which can prevent misunderstandings and increase interoperability on a semantic level

However, the main argument for exploring the topic of structured data input is that it has the potential to deliver completely accurate input already today. In addition, a direct correspondence between the manual input and the information model used by the input device greatly improves the conditions for accomplishing a computer based dialogue system.

In this paper we present a concept for structured human reporting based on an ontology-driven adaptable user-interface. The concept lays the foundation for the implementation of a possibly hand-held in-field reporting system, which can adapt to the context of the reporting situation as well as to possible information needs of other agents in the intelligence system. More specifically we put the following requirements on the system:

- It should be intuitive to a non-expert, who is neither an ontology engineer nor a domain expert.
- It should be domain independent, i.e. the system should work with ontologies from different domains.
- The output should be rdf-triples adhering to the ontology.
- It should be adaptable to the context of the reporting situation (who is reporting, what is the role of the reporter, where is the reporter, what time).
- It should be adaptable to the information needs of other agents in the intelligence system.

Fig. 1 gives an overview of how the system is intended to adapt to capture external information needs. The user observes an event and enters event information in the reporting system. The output of the reporting system is semantic statements. These statements are matched with information needs from other parts of the systems, which also are expressed as semantic statements. If there is a match, the information need is presented to the user as prioritized information to enter.

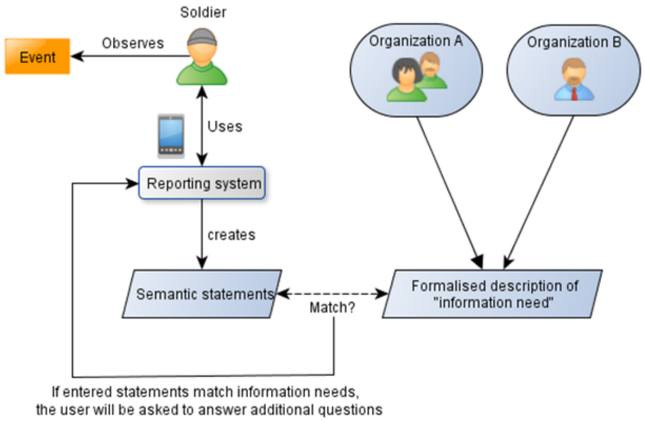

Soldier
Event
Observes
Uses
Organization A
Organization B
Reporting system
creates
Semantic statements
Match?
Formalised description of "information need"
If entered statements match information needs, the user will be asked to answer additional questions

Figure 1. An overview of the process for capturing external information needs.

II. Related Work

There is not much work reported on supporting manual input of semantic data (i.e. ontology instances). Standard ontology editors, such as Protégé, allow instance creation but require advanced user knowledge both regarding the domain and ontology engineering. The Disciple-RKF system [1] supports semantic user input through "knowledge elicitation scripts", which specifies natural language queries to be shown to the user and then how to process the user's answer semantically. This gives a good input support for a non-expert user, but requires an extensive manual work for the system engineers when defining the scripts as the logic of the GUI is defined there rather than in the ontology itself.

More effort has been put into developing user friendly systems for the querying of semantic repositories, although as stated in [2] the works are mainly for ontology engineers and not meant to assist domain experts or novice users. Semantic querying share common ground with semantic data input as it includes the creation of semantic statements, which are used as templates for matching the repository content. There are at least four approaches to support users in constructing semantic queries: natural language, controlled natural language, graphical editors and forms.

- Natural language query interfaces for semantic querying is a daunting task as it involves all issues related to natural language processing plus the additional constraint that the output must comply with a specific ontology. Its usability for querying large semantic web database is discussed in [3].

- Controlled natural language (CNL) defines a restricted form of natural language (e.g. English). It is used in a number of tools [4][5][6] developed for editing and querying ontologies. The disadvantage of CNL is that although the user can write and understand queries there is still an issue with learning the specific rules and boundaries of that particular CNL.

- Graphical ontology query tools are visual query systems that provide graphical notations to pictorially express semantic queries to retrieve data from semantic repositories. A number of scientific prototypes exist [2][7][8], which all however require the users to have knowledge about ontologies.

- The final approach for semantic query construction support is to use forms. In its simplest form it is just a predefined template, like an instance template in Protégé. More advanced support can include auto-completion, filtering and model checking [9].

In this paper we have due to the limitations of the other approaches chosen to build on the ideas of "smart" forms, extending them with more advanced methods for adaptation to context and external information needs.

III. Scenario

The following scenario illustrates the usage of the suggested system:

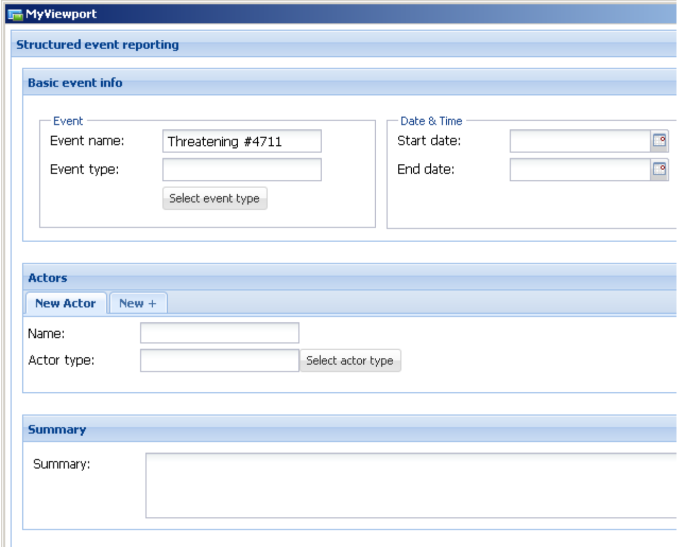
An army patrol is visiting a village. An officer of the patrol talks to the village leader who explains that the village was visited by a group of Talibans the week before. The village leader further describes the group as consisting of approximately 100-150 people and that they were threatening the population in order to get food.

The officer uses the reporting tool to enter information about the event. After manually choosing "threatening" as the main event type the tool automatically asks for related information, e.g. generic attributes as event "date" and "location", but also attributes and relationships specific to "threatening" like who is the "perpetrator" and "victim". The tool stores the information as triples in an rdf-repository. Once there, it is matched to external *requests for information* (RFIs) which have been posted by other people in the system. In this case there happens to be an RFI from the headquarter asking for information about what kind of weapons the Talibans possess. The statements of the report that our patrol officer is entering match this RFI as they are both about Talibans. The match triggers the reporting tool to present the RFI, so that the officer can make additional queries to the village leader.

IV. Conceptual Design

A. *Overview*

The overall idea of the reporting system is that it should adapt the interface based on what the user is reporting and take external information needs into consideration. In the event reporting scenario described above, the system should be loaded with a suitable military reporting ontology with attributes from e.g. the JC3IEDM. As an entry point the reporter is encouraged to report some basic event information consisting of the event type, time and place and information about the source (Fig. 2).



MyViewport

Structured event reporting

Basic event info

Event

Event name: Threatening #4711

Event type:

Date & Time

Start date:

End date:

Actors

Name:

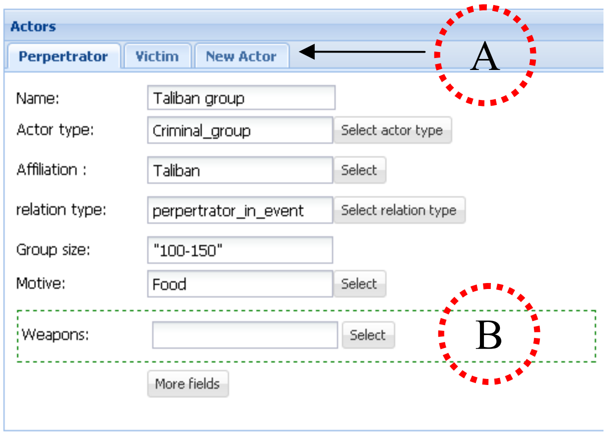
Actor type:

Summary

Summary:

Figure 2. Initially the interface only includes fields for basic event information.

Depending on what event type is chosen, new fields will emerge for the reporter to fill in. In the case of the Taliban scenario, the reporter chooses "threatening" as event type and will then be asked about which actors that were involved, there respective roles (perpetrator or victim) and additional properties that are related in the underlying ontology ("A" in Fig. 3)



Actors

Name: Taliban group

Actor type: Criminal_group

Affiliation : Taliban

relation type: perpertrator_in_event

Group size: "100-150"

Motive: Food

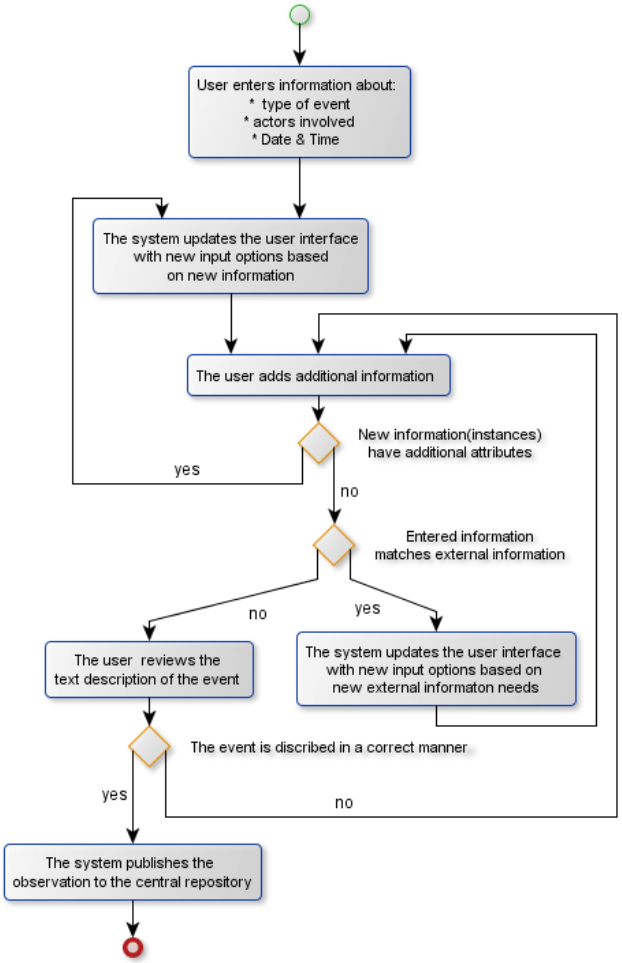
Weapons:

Figure 3. Depending on the user's choice of event type, related actor types emerge as new tabs (A). External information needs (B) emerge when entered information matches an RFI.

B. *Matching external information needs*

In addition to adapting the user interface by adding or removing input options based on what the user enters, the system will also match the event description with external information needs. In the Taliban scenario, an external information need had been registered in the form of an RFI, asking about the kind of weapons that the Talibans possess. The RFI is expressed as a set of semantic statements, which allows semantic matching. When the reporter enters affiliation "Taliban" for the perpetrator, this will trigger a match with the RFI. An additional field will emerge in the reporting tool asking for weapons information ("B" in Fig. 3).

A starting point is to match actors, places and event types between the event and external information need. If there is a match, the user might possess or have access to additional valuable information not reported yet. The matching process could also be done by executing a SPARQL query on the statements. If the result, with a degree of fuzziness, matches the information, the system asks the user some additional questions. A detailed description of the matching process is given in Fig. 4.



User enters information about:
* type of event
* actors involved
* Date & Time
The system updates the user interface with new input options based on new information
The user adds additional information
New information(instances) have additional attributes
yes
no
Entered information matches external information
no
yes
The user reviews the text description of the event
The system updates the user interface with new input options based on new external informaton needs
The event is discribed in a correct manner
yes
no
The system publishes the observation to the central repository

Figure 4. A detailed description of the matching process.

C. *Adaptable interface*

The ontology can be used to filter out irrelevant input fields and selection options. Besides type definitions, an ontology also defines relationship types and specifies when and how the relationships can be used. A relationship type can be restricted to only be valid from one kind of instance (domain) to another kind of instance type (range). Specifying domain and range provides means for creating a user interface with an increased level of usability since unsuitable input fields can be hidden. For instance, if the user wants to add a fact about an actor or an event, only the properties that have the corresponding domain will be accessible.

The available input fields can in our concept also be prioritized. In a time critical situation, it's important that the observer focus on what's important rather than trying to fill out all available fields. In a threat scenario, the victim's ethnicity may be a prioritized attribute to report, whereas in a crime investigating scenario, the shoe size may be a relevant attribute.

How the attributes are prioritized are scenario and context dependent. The priorities are also influenced by external RFI's. Consequently, the priorities are dynamic and the reporting system should be able to adapt to new priorities on the fly. In order to speed up the reporting, available contextual information should be used. This could mean automatically inserting information about time and place (by using GPS information).

Since we focus on using structured input fields which correspond to formally defined concepts we avoid using free text fields. By avoiding free text fields, there is a chance that the user thinks that the system didn't catch the meaning or some details. For this reason, the system will also provide a summary in natural language generated from the formal statements.

V. Discussion and Future Work

The tool presented in this paper is only a conceptual description. The next step is to do a proof of concept implementation and perform user tests. A setup for a thorough user evaluation could look like the following.

An ontology of a domain of interest is constructed together with a set of "observations" and a set of RFIs. The observations should consist of three parts:

- Part A contains the information that the test person should try to report, presented in either free text, or as an image or a combination.
- Part B contains additional information that the reporting agent has access to but don't enter unless someone asks for it. This could also be free text, an image or both.
- Part C contains the "correct" triples according to the test leader or some third party person/group. This part should not be revealed to the test person.

The RFIs should be in RDF-triples, where each RFI simulate the information need of another actor.

The test person is given the task to input the information presented in Part A of the observations. If the entered information matches the RFIs, the information from Part B can be used to answer any additional RFI related questions that the system presents to the test person. The resulting report is then compared to Part C and evaluated according to the following measures:

- the time to enter the information,
- the correctness of the resulting report,
- the completeness of the entered information, and
- the number of RFIs that were correctly answered.

Acknowledgment

This work was supported by the FOI research project "Tools for information management and analysis", which is funded by the R&D programme of the Swedish Armed Forces

References

[1] G. Tecuci, M. Boicu, D. Marcu, B. Stanescu, C. Boicu and J. Comello, "Training and Using Disciple Agents: A Case Study in the Military Center of Gravity Analysis Domain," in AI Magazine, 24,4, 2002, pp.51 - 68. AAAI Press, Menlo Park, California, 2002.

[2] A. Fadhil and V. Haarslev, "OntoVQL: A Graphical Query Language for OWL" Proceedings of the 2007 International Workshop on Description Logics (DL-2007), Brixen-Bressanone, Italy, June 2007, pp. 267-274.

[3] E. Kaufmann and A. Bernstein, "Evaluating the usability of natural language query languages and interfaces to Semantic Web knowledge bases," Web Semantics: Science, Services and Agents on the World Wide Web, vol. 8, no. 4, pp. 377 - 393, 2010.

[4] A. Funk, V. Tablan, K. Bontcheva, H. Cunningham, B. Davis, S. Handschuh, "CLOnE: Controlled Language for Ontology Editing", in Proceedings of 6th International Semantic Web Conference (ISWC), Busan Korea, November, 2007

[5] V. Tablan, T. Polajnar, H. Cunningham, K. Bontcheva, "User-friendly ontology authoring using a controlled language", in Proceedings of LREC 2006 - 5th International Conference on Language Resources and Evaluation. ELRA/ELDA, Paris, 2006

[6] R. Schwitter, K. Kaljurand, A. Cregan,C. Dolbear, and G. Hart, "A comparison of three controlled natural languages for OWL 1.1", in Proceedings of OWL: Experiences and Directions (OWLED 2008 DC), Washington, DC (metro), 2008.

[7] P. R. Smart, A. Russell, D. Braines, Y. Kalfoglou, J. Bao, and N. R. Shadbolt, "A Visual Approach to Semantic Query Design Using a Web-Based Graphical Query Designer," in Proceedings of the 16th international conference on Knowledge Engineering: Practice and Patterns, Berlin, Heidelberg, 2008, pp. 275–291.

[8] C. Kiefer and A. Bernstein, "The creation and evaluation of iSPARQL strategies for matchmaking," in Proceedings of the 5th European semantic web conference on The semantic web: research and applications, Berlin, Heidelberg, 2008, pp. 463–477.

[9] C. Mårtenson, A. Horndahl, "Using semantic technology in intelligence analysis", in Proceedings of Skövde Workshop on Information Fusion Topics, Skövde, Sweden, 2008.

.

Ontology-based Software for Generating Scenarios for Characterizing Searches for Nuclear Materials*

Richard C. Ward, Alexandre Sorokine, Bob Schlicher Michael Wright, Kara Kruse
Oak Ridge National Laboratory
Oak Ridge, TN 37831
wardrc1@ornl.gov

***Abstract*—A software environment was created in which ontologies are used to significantly expand the number and variety of scenarios for special nuclear materials (SNM) detection based on a set of simple generalized initial descriptions. A framework was built that combined advanced reasoning from ontologies with geographical and other data sources to generate a much larger list of specific detailed descriptions from a simple initial set of user-input variables. This presentation shows how basing the scenario generation on a process of inferencing from multiple ontologies, including a new SNM Detection Ontology (DO) combined with data extraction from geodatabases, provided the desired significant variability of scenarios for testing search algorithms, including unique combinations of variables not previously expected. The various components of the software environment and the resulting scenarios generated will be discussed.**

Keywords-component; ontology, software environment, scenario

I. Introduction

Recently there has been considerable interest in constructing computational systems that utilize ontologies in a multitude of ways [1, 2]. Examples are a semantic-based biosimulation modeling approach [3] that is being built on ontologies of anatomy and the physics of biology and the Gene Ontology (GO) [4] for bioinformatics. Here we present an ontology-based software framework for generating scenarios for a single searcher looking for the presence of special nuclear materials (SNM). Our software, the **ontology-driven scenario generator (ODSG),** will provide a capability to reason detailed scenario descriptions from limited user-input variables and create a multiplicity of scenarios with greater complexity than the initial input. The value to proliferation research is that this approach can be used to generate a wide variety of scenarios, incorporating complexities that were unobtainable from the intuitive heuristics, for testing detection algorithms.

The software system operates by first configuring an end-user application from the SNM Detection Ontology (SNM DO) and other data. Then the user selects scenario variables and ranges as desired. Once the variables are specified, a reverse process constructs the “data” for a series of scenarios using ontologies of data products and simulation models.

Each of the resulting scenarios can be viewed on the screen or encoded into XML or other formats, including KML [5], for further processing, and optionally converted into a human-readable narrative description. With the addition of building heights, elevations of floor levels, searcher, mobile objects, sources and other entities in the scene, the scenarios can be rendered using three-dimensional rendering software such as Blender [6].

II. General Assumptions

The present version of the ODSG software is intended to simulate an urban environment that is traversed by a single searcher on foot carrying a gamma-ray detector in a backpack. Each scenario is generated for an urban setting defined as an area in a city and described by a user-selected set of general descriptors. These general descriptors may include: location type (e.g., “city on the East coast”), the weather (temperature, humidity, etc.), information on the background radiation environment (e.g., possible presence of individuals treated with radioisotopes, presence of man-made objects, industry), hypothesized illicit locations of SNM source, and the general direction and walking time of the searcher carrying the detector.

Further, searching is assumed to be conducted only in the outdoor environment of the city with the searcher walking in non-adaptive patterns based on the shortest path to cross the search area; in this version the presence of a source does not alter the searcher’s path. The software design is flexible enough so that future versions could account for teams of searchers and adaptive searching with more complex search protocols.

III. Development of Supporting Ontololgies

ODSG uses multiple ontologies to infer from a general description (a list of user-input variables) to a much more complex detailed description and generates scenarios that are used later to test algorithms of SNM detection. We developed the SNM DO based on a multitude of sources including interviews of subject matter experts (SMEs), field manuals, textbooks, and other sources. SNM DO depicts an SNM detection environment the way it is perceived by the SMEs and outlines elements of the detection environment that may affect sensor readings in the opinion of the SMEs. Fig. 1 shows the general structure of the SNM DO.

In addition to the SNM DO, several other ontologies aimed at depicting the latent background knowledge, were developed. Overall ontology development methodology was based on Basic Formal Ontology (BFO) [7]. Several ontologies were developed that describe geographic data sources, such as TIGER [8], DHS Homeland Security Infrastructure Program (HSIP) [9], and others. Also we developed ontologies for simulation models, such as models for simulating paths of moving objects and pavement and sidewalk configurations. These simulation models were used during scenario generation to substitute for missing or unavailable data. SNM DO was matched with data source and model ontologies using an intermediate ontology based on the entries commonly found in the dataset ontologies and other geographic ontologies such as SWEET [10].

The ontologies were developed using the Simple Ontology Format (SOFT) [11] that provides such capabilities as visualization of ontologies in GraphViz and reasoning over a hierarchy of entities and relations [12]. An example SOFT diagram of portions of the SNM DO is shown in Fig. 2.

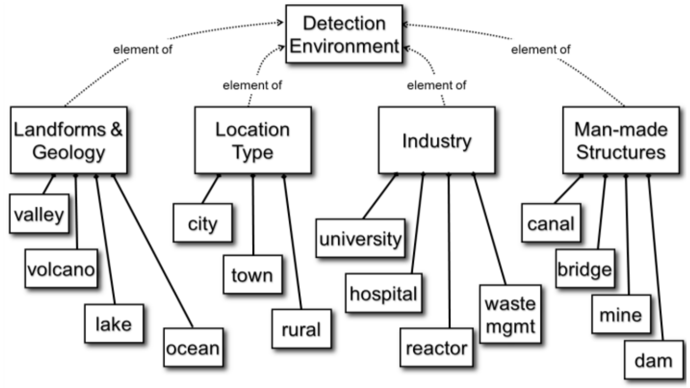


Figure 1. A portion of the special nuclear materials detection ontology (SNM DO). This portion focuses on geographic features.

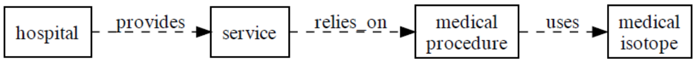


Figure 2. A portion of the SNM DO relating hospitals, procedures, services, medical procedures and isotopes.

IV. Software Architecture

ODSG system architecture is built around utilizing ontologies in various parts of its data processing cycle. The SNM DO and supporting ontologies are used to configure the interactive scenario generator. At the configuration stage entities from the ontologies are used to link the geodata sources and simulation models and generate the graphical user interface (GUI) of the end-user application. At the scenario generation stage user input is received through the GUI and used to construct the data by either retrieving it from the matching location in the geodatabase or running simulation models if such data are not available.

V. User Interface

A web interface was developed to capture the user's general input descriptors for the scenario. ODSG is a web-based application whose GUI is generated semi-automatically from the SNM DO and supporting ontologies. Each entity in the SNM DO corresponds to a scenario variable that can be controlled by the end-user through the GUI. Also each SNM DO entity is matched to a corresponding entity in the supporting ontologies that describes geographic datasets and simulation models available for scenario generation. The GUI generator uses these matches to deduce properties of an input variable such type (numeric, enumerated, geographic, etc.) and domain and appropriately formats the GUI elements of that variable.

The end user selects the variables of interest and provides ranges of values for those variables. For example, the user's selection might include geographic region, population, terrain type, presence of major buildings, roads, bridges, etc. associated with the location, and the presence of mobile objects such as people, cars, trucks, etc.

Given the user's input, the program selects a real urban location satisfying those criteria (e.g., East coast city with hospital and university near the scenario center). Following our assumption that the scene is restricted to the plan of the urban landscape provided by maps discussed above, a few of the variables governing scenario generation are: a) the path taken by the individual with the detector (allowed areas of walkable map); b) the types of shielding associated with the buildings or structures - these could be inferred using the ontology from the building type or use (government, school, store, etc.); c) characteristics such as types of soil, types of building materials commonly used, the vegetation present and weather (humidity or rain); d) the presence of or inference of known medical sources in individuals who have been treated or diagnosed using medical radioisotopes; and e) the presence of mobile objects such as cars, pedestrians, etc. Variation in the range of these variables comes partly from inferencing via the ontology and partly from random sampling over assumed typical ranges.

We also use ontologies to reason additional data from existing sources. For example, the possibility of finding anthropogenic radiation sources used in medical treatments can be inferred from the presence of the hospitals of certain types with the search area and thus the presence of treated individuals. Such radiation sources can be detected by the searcher. Fig. 2 illustrates one of these cases - if a hospital in the search area *provides* an oncology service that *relies_on* ventilation/perfusion (V/Q) procedures pulmonary perfusion (that *uses* Tc-99m) and pulmonary ventilation (that *uses* Xe-133), patients exiting this hospital might carry these specific medical isotopes. A SNM detection algorithm must be able to recognize these anthropogenic background sources.

VI. Mobile Objects

In the scenario generation we had to deal with mobile objects, entities such as the searcher, pedestrians, vehicles, etc. that move through the scene or otherwise change as a function of time. The searcher path is accomplished by weighting each point in a grid on the urban landscape, removing any points that have weights above a defined value (for example buildings, water features, etc.) that the searcher could not traverse, and creating an undirected graph. Using the A*search algorithm [13] a path is computed through this landscape of weighted values that is the minimum path between arbitrarily selected endpoints on roads at the edge of the scene. In addition, we used MASON [14], open source agent-based modeling (ABM) tools, to update and track the objects as they moved through the scene. The ODSG software provides random paths for up to ten pedestrians and ten vehicles.

VII. Geographical Information Systems

ODSG is a web application that uses a PostgreSQL [15] database with the PostGIS extension [16] for most of its storage and data processing needs and Minnesota MapServer [17] for geographic display of the resulting scenarios. Using GIS the track of any mobile object (searcher, pedestrian, vehicles, etc.) is easily visualized and correlated to the text narrative. The approach of combining inferencing from ontologies within a GIS framework to generate scenarios enhances the capability to generate and visualize scenarios for evaluation of SNM detection algorithms. The scenarios were passed to a narrative generator where they are converted into English sentences. In addition, they can be delivered in XML format which could be passed to a 3D-georenderer (Blender) for three-dimensional display of the scene [6], or KML format to be viewed in Google Earth.

VIII. Results

During the system demonstration, ODSG was used to generate about a hundred scenarios using several sets of input variables. In many cases multiple scenarios were generated from the same input data set by using iteration over the permitted ranges of variable values. All scenarios had a single searcher in the scene and many had pedestrians and/or vehicles in the scene, demonstrating the capability of adding mobile objects to the scenario generation.

An example of the capability to generate multiple scenarios from a single input is the sixteen scenarios created from the user input shown in Table I. The user input for “General US Region” is New England. The user has also selected presence in or near the scene of a railway and a port. The GIS map for one of the sixteen scenarios generated (sc0126_005) is shown in Fig. 3. Each scenario displays the searcher path (dark circles) as well the track of three vehicles (squares) passing through the scene. The railway is seen in the bottom portion of Fig. 3. The combination of location and presence of various infrastructures (such as railways and ports) generates multiple output scenarios. This example demonstrates the ease with which a large set of detailed scenarios can be constructed from a much simpler set of generalized user input variables.

Table I. A Portion of the User Input Variables for Example

Number of searchers	1		
Number of pedestrian	0		
Number of vehicles	3		
General US Region	New England		
Type of Detector	Handheld	**Material**	$LaBr_3$
Type of Search	Event-driven	By protocol	
Near search area	Railway	Port	

IX. Conclusions

Utilizing both domain-specific ontologies and those containing latent-background terminology, we have created a software environment that generates an expanded number of scenarios from a general set of user input variables for purposes of testing algorithms for detection of SNM. The specific ontology developed, the SNM DO, was built using subject-matter expert knowledge of the detection process for searchers on foot in an urban setting. The detailed dependence of the software construction and operation on the ontologies is described and a specific example of the user input variables used to create sixteen scenarios is elaborated. By using ontologies both to configure the software architecture and to drive inferencing based on ontological reasoning, we greatly expanded the number and variety of scenarios generated from a single set of user input. Such applications show the importance of incorporating ontologies into software frameworks for generation of scenarios for activities such as searching for nuclear materials.

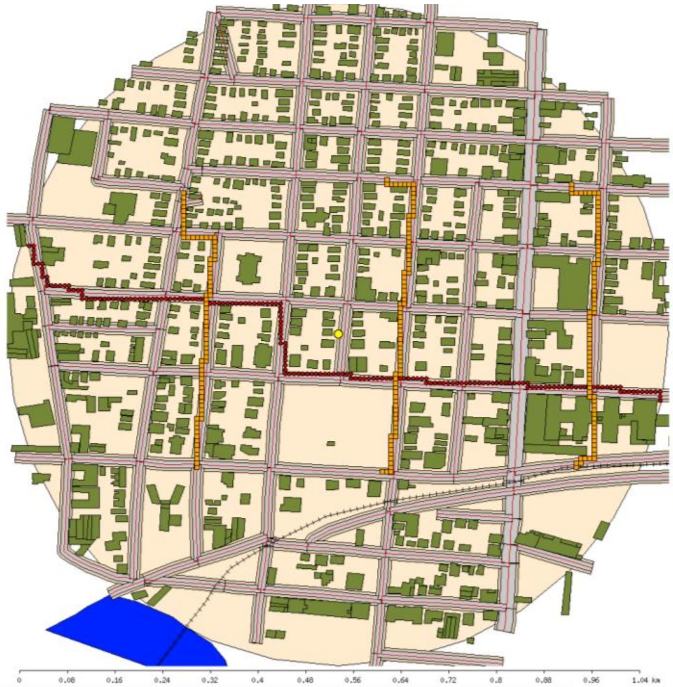


Figure 3. Scenario sc0126_005 generated from input in Table 1. The searcher path is shown with dark circles and the vehicle tracks with squares.

ACKNOWLEDGEMENTS

* Research sponsored by U.S. Department of Energy/National Nuclear Security Administration NA-22 Simulation, Algorithms and Modeling Program under contract: OR10-Ontology Demo-PD06.

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

REFERENCES

[1] Klein, M., D. Fensel, D. van Harmelen, F, and Horrocks, I., The relation between ontologies and XML schemas**,** *Linkoping Electronic Articles in Computer and Information Science*, 2001.

[2] Quix, C., Kensche**,** D.**,** Li, X. (2007) Matching of ontologies with XML schemas using a generic metamode**l**, Lecture Notes In Computer Science, Proceedings of the 2007 OTM Confederated international conference on the move to meaningful internet systems, Vilamoura, Portugal, 2007, Pages: 1081-1098

[3] Neal, M., (2010) Modular semantics-based composition of biosimulation models. Ph.D. Thesis, University of Washington.

[4] Gene Ontology (GO) http://www.geneontology.org/, 2011.

[5] Keyhole Markup Language (KML) http://code.google.com/apis/kml/documentation, 2011.

[6] Roberts, R and P. Pope, private conversations, 2011.

[7] Spear, A. D. (2006) *Ontology for the Twenty First Century: An Introduction with Recommendations*. Saarbrücken, Germany: IFOMIS. Retrieved from http://www.ifomis.org/bfo/documents/manual.pdf

[8] Topologically Integrated Geographic Encoding and Referencing system *(TIGER) http://www.census.gov/geo/www/tiger/*, 2011.

[9] Homeland Security Infrastructure Protection (HSIP) Gold 2011. The Department of Defense, National Geospatial-Intelligence Agency, Office of Americas, North American and Homeland Defense Division, 2011.

[10] Semantic Web for Earth and Environmental Terminology (SWEET) Ontologies, http://sweet.jpl.nasa.gov/ontology/, 2011.

[11] Simple Ontology FormaT (SOFT) http://github.com/sorokine/SOFT (in preparation).

[12] Gansner, E. R., & North, S. C. (2000). An open graph visualization system and its applications to software engineering. *Software – Practice and Experience* **30** (11), 1203--1233.

[13] Hart, P. E.; Nilsson, N. J.; Raphael, B. (1968). "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". *IEEE Transactions on Systems Science and Cybernetics SSC4* **4** (2): 100–107

[14] MASON: A fast discrete-event, multi-agent simulation library, http://cs.gmu.edu/~eclab/projects/mason/.

[15] PostgreSQL: The world's most advanced open source database. (n.d.). Retrieved October 20, 2011, from http://www.postgresql.org/

[16] Team, P. D. (2007). *PostGIS – Spatial extension to PostgreSQL*. http://postgis.refractions.net/.

[17] MapServer, http://mapserver.org/, 2011.

Use of Ontology to Facilitate the Creation of Synthetic Imagery of Industrial Facilities

Paul Pope
Los Alamos National Laboratory
Los Alamos, NM, USA
papope@lanl.gov

Randy Roberts
Lawrence Livermore National Laboratory
Livermore, CA, USA
roberts38@llnl.gov

Abstract—**Algorithms which perform auto-annotation of remotely sensed imagery need to undergo verification and validation (V&V) such that the end user can make a fitness-for-use judgment regarding their particular application and can be assured of a high level of confidence in achieving success. Synthesizing these data is one means of obtaining the imagery required to conduct benchmark testing. This paper presents a system to create benchmark imagery of industrial facilities for conducting V&V of auto-annotation algorithms. The method proposes to leverage an ontology of industrial facilities to capture domain knowledge regarding both the industrial process flow as well as the objects required to support the industrial process at a particular production level.**

Keywords-verification and validation; benchmark imagery; industrial facility; synthetic image

I. Background

The recent rise in collection of remotely sensed imagery of the Earth is driving the need for automated means to process these data to extract important information for addressing a variety of civilian and intelligence problems. One problem to be addressed is the detection, identification, characterization, and monitoring of industrial facilities. Auto-annotation algorithms are being developed which strive to meet this need [1]. An important step in the development of such auto-annotation algorithms is a verification and validation (V&V) strategy [2]. A properly designed and implemented V&V strategy establishes and quantifies the conditions under which an auto-annotation algorithm can be applied to imagery with an expectation of success. Furthermore, a key component of the V&V methodology is a large, well-designed set of benchmark imagery [3], [4]. Due to the large number of extrinsic factors and their levels which must be provided for (e.g., various view angles, times of day, seasons, backgrounds, etc.), and the resulting combinatorial explosion, creation of realistic synthetic imagery must be considered as a means to obtain the required number and variety of benchmark imagery for conducting V&V [5].

Herein we propose an approach to synthesizing benchmark imagery of industrial facilities. Achieving realism means more than photo-realism. The facility layout must truly represent the actual process flow of a real industrial process, as well as the object types, sizes, and number required to meet a particular level of production capacity. Therefore, central to our approach is an application-level ontology that provides a principled means to organize the various types of industrial facilities and to determine the objects which compose a particular facility. Our review of the relevant literature indicates that while work is beginning in the use of ontologies for auto-annotation of imagery (e.g., [6]), very little work has been conducted to date on the use of ontologies to synthesize the imagery required to conduct V&V of such algorithms.

II. Synthetic Image Creation

The proposed system is described here and illustrated in Fig. 1. The process would be initiated by the user defining the type of industry to be modeled (e.g., aluminum smelting), and the production rate (e.g., 175 kilotons per year) [7]. Extrinsic parameters (e.g., view angle, time of day, season, clutter, etc.) would also be defined at this point. Setting the type of industry would queue the system to select the associated process flow from a process flow database. The process flows in this database would be stored as networks (e.g., linked-list trees). The nodes of the process flow networks would set the type of objects required to conduct the process (e.g., tanks) and the object's use (e.g., storage). The desired production rate would drive the sizing and number of these objects. Since these characteristics are interrelated, a structural engineering database would provide limits on the realistic minimum and maximum dimensions allowed for each object. These limits would resolve the ambiguity in the number of objects required to provide the storage capacity necessary to support the desired production rate, without violating structural engineering constraints.

The process flow, required objects, and their size and number would then be used in a facility layout algorithm to arrange and orient all the objects. A spatial topology might be enforced, formulated through a cost minimization criterion [8], or it could be statistical in nature [9].

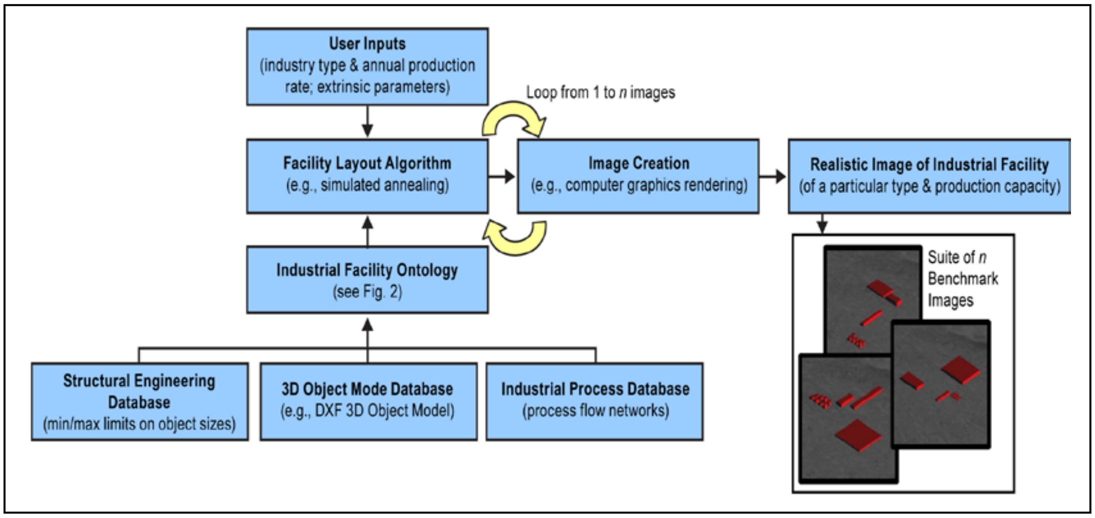

User Inputs
(industry type & annual production rate; extrinsic parameters)
Loop from 1 to n images
Facility Layout Algorithm
(e.g., simulated annealing)
Image Creation
(e.g., computer graphics rendering)
Realistic Image of Industrial Facility
(of a particular type & production capacity)
Industrial Facility Ontology
(see Fig. 2)
Suite of n Benchmark Images
Structural Engineering Database
(min/max limits on object sizes)
3D Object Mode Database
(e.g., DXF 3D Object Model)
Industrial Process Database
(process flow networks)

Figure 1. Illustration of the proposed system to synthesize imagery of industrial facilities.

It is possible for a multitude of layouts to be generated, even though the process flow and the type, number, and size of objects remains the same. This means that variation in facility layout is provided at this point in the process.

Therefore, a for loop is utilized such that a number of images can be output while still holding fixed the type of industry and its annual production output.

Once the object arrangement has been computed, an image of the industrial facility is created via rendering, either through a physics-based method [10] or through computer graphics methods [11]. On exit from the loop over the number of images desired, the required suite of benchmark imagery will have been produced.

III. Industrial Facility Ontologies

Ontologies would be leveraged at two places within this process framework (Fig. 2). First, the industry type would be selected from an ontology of industrial types (top half of Fig. 2). Second, the object types would be selected from an ontology of industrial process object types (bottom half of Fig. 2). These ontologies would either be created by information gleaned from subject matter experts via knowledge elicitation and a review of the relevant literature, or leveraged from existing ontologies, or a combination of both [12]. An initial review of ontologies which capture industrial processes reveals that they appear to be quite specialized and are generally rare. Examples are the MAnufacturing Semantics Ontology (MASON) [13] and OntoCAPE [14]. Creation of an ontology designed for our particular purpose (i.e., containing only the objects which are "relevant" within our "reality") will most likely be required [15]. Also, considering the fact that we will have to account for industrial parts and wholes, their spatial relations, as well as geographic "things", then insights into mereotopology [16] and geo-ontology [17] will most likely be required and should prove useful.

IV. Conclusions and Future work

A system to create synthetic imagery of industrial facilities for the purpose of conducting V&V of auto-annotation algorithms has been proposed herein. Central to our design is an industrial facility ontology which guides the selection of the object types and their number to re-create the industrial process desired and its production rate.

Realism is achieved both by leveraging the industrial facility expertise captured by the ontology as well as the impressive realism available via modern computer graphics techniques and technology.

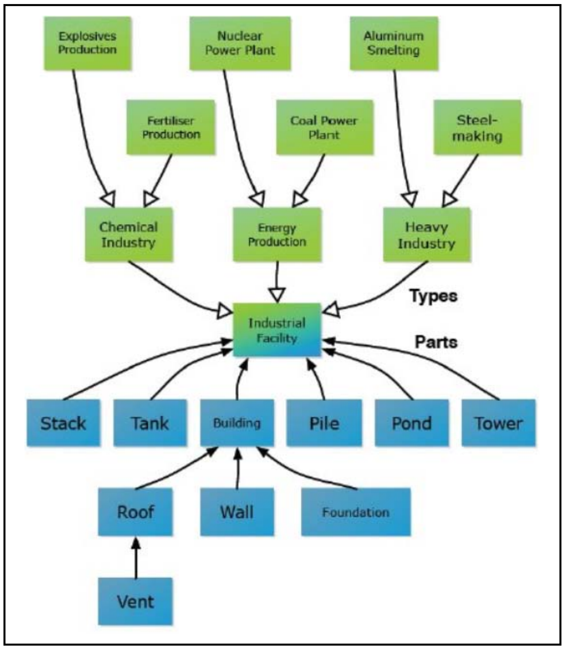


Figure 2. Illustration of an industrial facility ontology to support the proposed system. The upper relationships indicate industry types, while the lower relationships indicate parts (objects) that comprise an industrial facility. This ontology was derived in part by analysis of the nouns put forth as salient by Chisnell and Cole [18].

This overall sketch is an important first step in achieving such a capability; however, much work remains to be done. Our current aim is to realize a first version of such a system. We expect that substantial improvements will occur as this nascent version is utilized for V&V of auto-annotation algorithms.

Acknowledgment

The authors extend their sincere thanks to the program office of Simulations, Algorithms, and Modeling (SAM), of the National Nuclear Security Administration's Office of Nonproliferation Research & Development, for support of this research. The authors also extend their thanks to both Ian Burns (LANL) for the 3D synthetic images of a small chemical facility used in Fig. 1 and to Raju Vatsavai (ORNL) for the ontology illustration used in Fig. 2.

References

[1] S. Gleason, M. Dema, H. Ferrell, A. Cheriyadat, R. Vatsavai, and R. Ferrell, "Verification & Validation of a Semantic Image Tagging Framework via Generation of Geospatial Imagery Ground Truth," in Proc. IGARSS 2011, July 24-29, Vancouver, Canada, in press.

[2] R. Roberts et al., "On the verification and validation of geospatial image analysis algorithms," in Proc. 2010 IEEE Int'l Geoscience and Remote Sensing Symposium, July 2010, pp. 174–177.

[3] T.L. Berg et al., "It's all about the data," Proceedings of the IEEE, vol. 98, no. 8, pp. 1434–1452, August 2010.

[4] J.A. Shufelt, "Performance evaluation and analysis of monocular building extraction from aerial imagery," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 21, no. 4, pp. 311–326, April 1999.

[5] R. Roberts et al., "Design of benchmark imagery for validating facility annotation algorithms," in Proc. IGARSS 2011, July 24-29, Vancouver, Canada, in press.

[6] N. Durand, S. Derivaux, G. Forestier, C. Wemmert, and P. Gancarski, "Ontology-based object recognition for remote sensing image interpretation," in Proc. 19th IEEE International Conference, on Tools with Artificial Intelligence pp. 472-479.

[7] "List of Aluminum Smelters," http://en.wikipedia.org/wiki/List_of_aluminium_smelters, last accessed September 16, 2011.

[8] F. Richard, et al., *Facility Layout and Location: An Analytical Approach*, Prentice Hall, Second Edition, 592 pp.

[9] L. Chwif, M. Barretto, and L. Moscato, "A Solution to the Facility Layout Problem using Simulated Annealing," Computers in Industry, vol. 36, 1998, pp. 125-132.

[10] J. Mason et al., "Validation of contrast and phenomenology in the Digital Imaging and Remote Sensing (DIRS) lab's image generation (DIRSIG) model," 1994, vol. 2269, pp. 622–633, SPIE.

[11] J. Foley, A. van Dam, S. Feiner, and J. Hughes, *Computer Graphics: Principles and Practice*, Addison-Wesley Professional, Third Edition, 2012, 1472 pp., in press.

[12] Mizoguchii, R. "Part 1: Introduction to Ontological Engineering," in *Tutorial on Ontological Engineering*, New Generation Computing, Ohmsha, Ltd. and Springer-Verlag, 2003, pp. 365-384.

[13] S. Lemaignan, A. Siadat, JY Dantan, and A. Semenenko, "MASON: A Proposal for an Ontology of Manufacturing Domain," in Proc. IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06), 6 pp.

[14] J. Morbach, A. Yang, and W. Marquardt, "OntoCAPE – A Large-Scale Ontology for Chemical Process Engineering," Engineering Applications of Aritificial Intelligence, vol. 20, pp. 147-161, 2007.

[15] Ceusters, W., "Towards a Realism-Based Metric for Quality Assurance in Ontology Matching," Proc. of the International Conference on Formal Ontology in Information Systems, (FOIS 2006), Baltimore, Maryland, 9-11 November, 2006.

[16] Smith, B.W., "Mereotopology: A Theory of Parts and Boundaries," *Data and Knowledge Engineering*, Vol. 20, 1996, pp. 287-303.

[17] NGA, "Geospatial Ontology Trade Study," final report prepared by Northrop Grumman Corporation Information Technology – TASC, jointly with BBN Technologies, June 27, 2007, 46pp.

[18] T. Chisnell and G. Cole, "Industrial Components—A Photo Interpretation Key on Industry," Photogrammetric Engineering, vol. 24, pp. 590–602, March 1958.