SEMANTIC TECHNOLOGY FOR INTELLIGENCE, DEFENSE AND SECURITY

STIDS 2010

Setting the Stage for High-level Knowledge Fusion

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

Conference held at:

The Mason Inn
George Mason University
Fairfax, Virginia Campus
27 - 28 October 2010

Online Proceedings

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

STIDS 2010

Preface

STIDS 2010 provided a forum for academia, government and industry to share the latest research on semantic technology for defense, intelligence and security applications. Formerly known as Ontology for the Intelligence Community (OIC), the conference was broadened in scope to encompass a larger range of problems in defense, intelligence and security. Semantic technology is a fundamental enabler to achieve greater flexibility, precision, timeliness and automation of analysis and response to rapidly evolving threats.

The conference was an opportunity for collaboration and cross-fertilization between researchers and practitioners of semantic-based technologies with particular experience in the problems facing the Intelligence, Defense, and Security communities. It will featured invited talks from prominent ontologists and recognized leaders from the target application domains.

To facilitate interchange among communities with a clear commonality of interest but little history of interaction, STIDS hosted two separate tracks. The Research Track showcased original, significant research on semantic technologies applicable to problems in intelligence, defense or security. The Applications Track provided a forum for presenting implemented semantic-based applications to intelligence, defense, or security, as well as to discuss and evaluate the use of semantic techniques in these areas.

Fairfax, VA, October 2010.

Paulo Costa and Kathryn Laskey
STIDS 2010 Chairs

CONFERENCE CO-CHAIRS

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

SCIENTIFIC COMMITTEE

Name	Affiliation
Werner Ceusters	University at Buffalo
Paulo Costa	George Mason University
Tim Darr	Knowledge Based Systems, Inc.
Katherine Goodier	NCI, Inc.
Kathleen Stewart Hornsby	University of Iowa
Terry Janssen	Lockheed Martin Co. (SC Chair)
Kathryn Blackmond Laskey	George Mason University
Robert Latiff	George Mason University
Nancy Lawler	US Department of Defense
Kevin Lynch	CIA
Dan Maxwell	KaDSci, Inc.
Leo Obrst	MITRE Corporation
Robert Schrag	Global Infotek, Inc.
David Schum	George Mason University
Marvin Simpson	Optech, Inc.
Barry Smith	NCOR, University at Buffalo
Gheorghe Tecuci	George Mason University
Andreas Tolk	Old Dominion University
Duminda Wijesekera	George Mason University
Syed Abbas Zaidi	George Mason University

Research Papers

Maintaining Temporal Perspective

Ian Emmons and Douglas Reid

Raytheon BBN Technologies, Arlington, VA 22209, USA
{iemmons,dreid}@bbn.com

Abstract. We present methods for annotating data with the time when it was learned and for answering queries according to what was known at any point in time. Specifically, we present an RDF knowledge representation that associates facts with their transaction times, and a query mechanism that transforms a time-agnostic SPARQL query and a point in time into a new, time-sensitive query. The transformed query yields the subset of the results of the original query that were valid at the indicated time. In addition, the methods presented here enable non-destructive merging of coreferences. These techniques apply broadly to storage and retrieval systems that require time-based versioning of data and are essential for maintaining temporal perspective in rapidly-evolving analytical environments.

1 Background

There is a large body of work in the theory and construction of temporal databases, as summarized in [4]. This paper describes an application of that body of research to support the development of an operational, temporally-annotated *semantic* database.

The solution presented here grew from a project to develop a risk analysis application for assessing risks to one particular high-value resource. This system continually gathers data from five relational databases (non-temporal) and stores it as RDF in a triple store. The data includes events as well as latest current state, both of which are time-sensitive.

Analysts use the application to perform daily risk assessments, and these results are also placed in the triple store. Later review of these analyses is an important part of the analysts' work. This leads directly to the time-validity requirement: the triple store must maintain the temporal perspective of the data for subsequent review and inspection, because older analyses mean little in the context of current data. Note that temporal perspective may also be useful in the analysis task itself. For instance, the age of what we know and the order in which we learned it sometimes affect the interpretation.

The solution consists of two primary components. First is the RDF knowledge representation, which associates facts with the time intervals in which they were known (traditionally referred to as the *transaction time* of a given fact[4]). The second is the query rewriter, which transforms a time-agnostic SPARQL query and a point in time into a new, time-sensitive query. The transformed query

yields the subset of the results of the original query that were known at the given time.

This solution aims to track when facts were considered true. For that purpose, we have chosen a linear, discrete temporal model (defined in [4]) that deals with transaction times of facts. It is not intended to manage the explicit temporal aspects of data, such as occurrence times of events. In this way, this solution may be thought of as a form of provenance tracking.

2 The Knowledge Representation

We describe the knowledge representation through an example in which three data imports occur (see the summary timeline shown in Table 1). The first import yields the following statements:

```
:Person1 a :Person ;
  :name "Robert Jones" ;
  :ssn "123-45-6789" .
```

These describe a person with name and social security number (SSN). Identifiers preceded by colons are URIs whose base URI has been suppressed for brevity. To this the system adds a proxy consisting of the following statements:

```
:Proxy1 a :Individual ;
  :hasPrimitive :Person1 ;
  :usesValue [ :Person1 :name "Robert Jones" ] ,
             [ :Person1 :ssn "123-45-6789" ] ;
  :temporalIndex :TmpIdx1 .
:TmpIdx1 a :KbProperInterval ;
  :startedBy :Time1 ;
  :finishedBy :EndOfTime .
:Time1 a :DateTimeInterval ; :xsdDateTime "2009-08-17T00:00:00" .
:EndOfTime a :DateTimeInterval ; :xsdDateTime "9999-12-31T23:59:59" .
```

Proxies represent the sum total of information known about an entity for a specified time interval within our system. Borrowing from situation calculus, the proxies provide a mechanism for encoding the history of knowledge (or situation) and for resolving the truth values of properties (or fluents) of entities throughout that history[3]. From a philosophical perspective (i.e., BFO[5]), proxies can be viewed as Processual Entities that capture the time-specific Qualities and Realizable Entities of a particular Independent Continuant within our system.

The example proxy is of type `:Individual`, one of two subclasses of `:Proxy`, and points to the person entity via the `hasPrimitive` property. The choice of terminology "primitive" here will make more sense when we discuss coreference resolution below. The `usesValue` properties point to the attribute values of the person that are known during the time period of this proxy. Note that they point not to the objects of the person attribute statements, but rather to reifications of those statements.

There is no `usesValue` for the type of the person entity. This reflects a conscious design decision to avoid introducing time dependence into the RDFS inference performed by our triple store, greatly simplifying the implementation. One consequence of this decision is a restriction on the ontology: Classes must not carry time-dependent meaning. For instance, `:Person` is a perfectly reasonable class, but introducing a subclass of it like `:CEO` would be a mistake, because a person who is a CEO holds that position for only a portion of their life.

The remainder of the proxy, from the `:temporalIndex` property on, encodes the proxy's transaction time. This extends from the time of import (the opening second of August 17, 2009 in this case) until the end of time. If the representation seems more complex than necessary, this is because it must comply with the combined requirements of the OWL-Time ontology[1] and of the temporal index associated with our triple store,[2] Parliament™[2]. The temporal index, based upon Allen's Interval Algebra[1], allows us to query efficiently for such things as time intervals containing, intersecting, before, or after a given time interval.

Now suppose a second import, from a different data source the next day, yields this data:

```
:Person2 a :Person ;
  :name "Bob Jones" ;
  :ssn "123-45-6789" .
```

Due to the matching SSN and the similar names, most would say that these two data entities are "obviously" the same real-world entity, in other words that they form a coreference that we want to resolve, or merge, into a single entity. This is a common problem with multi-source data. It often arises simply because there is no universal system of unique identifiers, but it might also happen when entities are viewed from different domains. For instance, a bridge can be viewed as a transportation resource, a maintenance responsibility, or a target.

When we merge the entities of a coreference, there are two non-obvious but important requirements. First, merging should be reversible and non-destructive, and second we must maintain temporal perspective. To accomplish this, we first "retire" the original proxy created after the first import, which simply means that we delete the single `:finishedBy` statement and add a new one so that the proxy's transaction time is a closed interval:

```
:Proxy1 a :Individual ;
  :hasPrimitive :Person1 ;
  :usesValue [ :Person1 :name "Robert Jones" ] ,
             [ :Person1 :ssn "123-45-6789" ] ;
  :temporalIndex :TmpIdx1 .
:TmpIdx1 a :KbProperInterval ;
  :startedBy :Time1 ;
  :finishedBy :Time2 .
:Time1 a :DateTimeInterval ; :xsdDateTime "2009-08-17T00:00:00" .
:Time2 a :DateTimeInterval ; :xsdDateTime "2009-08-17T23:59:59" .
```

[1] http://www.w3.org/TR/owl-time/
[2] http://parliament.semwebcentral.org/

Then we add a second proxy like so:

```
:Proxy2 a :Merge ;
  :hasPrimitive :Person1, :Person2 ;
  :usesValue [ :Person1 :name "Robert Jones" ] ,
             [ :Person2 :name "Bob Jones" ] ,
             [ :Person1 :ssn "123-45-6789" ] ;
  :temporalIndex [ "2009-08-18T00:00:00" .. "9999-12-31T23:59:59" ] .
```

Here we have abbreviated the `:temporalIndex` for brevity. This proxy is of type `:Merge`, the other subclass of `:Proxy`, and has two primitives, namely both of the `:Person` entities imported so far. The `:usesValue` statements call out both of the names and one of the SSNs. (The other SSN is left out because it has the same value.)

Both `:Proxy1` and `:Proxy2` exist in the triple store at this point, and they have disjoint time intervals. This allows us to choose the appropriate proxy for any given point in time and then look up the corresponding state of the associated entity. Prior to August 17, 2009, there is no proxy and so this person is unknown. During August 17, 2009, the first proxy calls out just one name, and after that day the second proxy calls out both names. In addition, the proxy has effectively merged the coreference without changing the original two entities.

Now suppose that a third import from the original data source happens at 9:35:20 Zulu time on August 18, 2009, and that this re-imports the same "Robert Jones" record that we saw in our first import. However, suppose that in the interim the SSN was changed to correct a typo:

```
:Person1 a :Person ;
  :name "Robert Jones" ;
  :ssn "123-45-6789" , "123-45-6798" .
```

Importing the same record from the same database results in the same `:Person1` entity, since the import process forms the URI from the primary key of the record, but it creates a new `:ssn` property value, such that `:Person1` now has two SSN values associated with it. One comes from the third import itself, and the other is left over from the first import. Naturally, we now need to expire the second proxy and create new ones:

```
:Proxy2 a :Merge ;
  :hasPrimitive :Person1, :Person2 ;
  :usesValue [ :Person1 :name "Robert Jones" ] ,
             [ :Person2 :name "Bob Jones" ] ,
             [ :Person1 :ssn "123-45-6789" ] ;
  :temporalIndex [ "2009-08-18T00:00:00" .. "2009-08-18T09:35:19" ] .

:Proxy3 a :Individual ;
  :hasPrimitive :Person1 ;
  :usesValue [ :Person1 :name "Robert Jones" ] ,
             [ :Person1 :ssn "123-45-6798" ] ;
  :temporalIndex [ "2009-08-18T09:35:20" .. "9999-12-31T23:59:59" ] .
```

```
:Proxy4 a :Individual ;
  :hasPrimitive :Person2 ;
  :usesValue [ :Person2 :name "Bob Jones" ] ,
             [ :Person2 :ssn "123-45-6789" ] ;
  :temporalIndex [ "2009-08-18T09:35:20" .. "9999-12-31T23:59:59" ] .
```

There are two new proxies because the new SSN indicates that these two entities most likely do not represent the same person after all. The proxy for `:Person1` has `:usesValue` properties for the name and for the new `:ssn`, but not for the old `:ssn` value. Thus just before the third import, `:Proxy2` is valid and we see a single entity with two names and SSN, but just after the import we see two distinct entities with different names and SSNs.

Table 1. Summary timeline for example scenario illustrating knowledge store evolution

Date and Time	Actions
2009-08-17 00:00:00	1. `:Person1` added to the KB 2. `:Proxy1` created for `:Person1` and added to the KB
2009-08-18 00:00:00	1. `:Person2` added to the KB 2. `:Person2` discovered to be same person as `:Person1` 3. `:Proxy2` created to merge `:Person1` and `:Person2` 4. `:Proxy2` added to the KB 5. `:Proxy1` retired
2009-08-18 09:35:20	1. `:Person1` re-imported with a different SSN value 2. `:Person1` and `:Person2` un-merged by Analyst 3. `:Proxy3` created for `:Person1` going forward 4. `:Proxy4` created for `:Person2` going forward 5. `:Proxy2` retired

3 Query Rewriting

Writing time-sensitive queries according to the knowledge representation scheme can be a complex, error-prone, and tedious chore. To alleviate the burden of composing temporally-annotated queries, a query rewriting service was developed. This service automatically transforms a time-agnostic SPARQL query and a provided time into a new, time-sensitive SPARQL query. The resultant query yields a subset of the results from the initial time-agnostic query that are considered valid for the submitted time.

The rewriting service does not alter the meaning of the original query bindings to completely obscure the existence of proxies (merges and individuals) within the system. Rather, the service leaves the original query bindings in-place and adds variables to the result set to represent entity proxies. This behavior was

requested by our customer, as they wanted direct access to the unproxied entities in the query results. The alternative is to alter the meanings of the binding variables to refer to the proxies and not return the primitive entities themselves.

Query rewriting takes place in three distinct phases. First, we make an exact copy of the original query. Second, proxy representations are appended to the original query to match the underlying knowledge representation, appropriately following the structure of the submitted query. Finally, temporal selection information is appended for each proxy added during the second step.

To demonstrate the query rewriting service, consider this example:

```
SELECT DISTINCT ?person ?name
WHERE {
   ?person a :Person ; :ssn "123-45-6789" .
   OPTIONAL { ?person :name ?name . }
}
```

When submitted with the time `2007-08-17T12:00:00` to the query rewriting service, the resulting time-sensitive query is:

```
SELECT DISTINCT ?person_proxy ?person ?name
WHERE {
   ?person a :Person ; :ssn "123-45-6789" .
   ?person_proxy a :Proxy ;
      :hasPrimitive ?person .
      :usesValue [ rdf:predicate :ssn ;
                   rdf:object "123-45-6789" ] .
   OPTIONAL {
      ?person :name ?name .
      ?person_proxy :usesValue [ rdf:predicate :name ;
                                 rdf:object ?name ] .
   }

   ?person_proxy :temporalIndex ?interval1 .
   ?interval1 a :ProperInterval ;
       :intervalContains [ a :DateTimeInterval ;
                 :xsdDateTime "2009-08-17T12:00:00Z"^^xsd:dateTime ] .
}
```

The primary reason for copying the original query exactly is that it improves the baseline query performance of the modified translated query by providing optimization hints to our system's query optimizer. Our knowledge representation scheme relies heavily on statement reification to encode temporal validity. During development, we discovered that our query optimizer is easily confused by reification, often producing inefficient query clause orderings that result in time-prohibitive query executions. Leaving the original clauses in place essentially enables our optimizer to ignore the reified statements without altering the validity of the query results. Retaining the original query structure also improves readability, which can be invaluable for system development and testing.

Proxy expansion closely mirrors the knowledge representation scheme for temporal annotation, with some noteworthy exceptions. Type clauses were ig-

nored in the proxy expansion step, as type information is considered invariant. In transforming query clauses to match the knowledge representation scheme, individual property clauses of a given entity are expanded to match the `:usesValue` construction of the knowledge representation scheme. However, the `rdf:subject` component of statement reification is dropped. This enables multiple underlying primitive entities to provide clause matches when the proxied entity is a Merge proxy in the knowledge store.

To add the proxy representation to the original query automatically, the query rewriting system relies upon the underlying domain ontology for hints about which query variables (and clauses) refer to entities (and properties) that require the introduction of temporal sensitivity. Proxy representations are only added for each entity that appears as the subject in a triple clause that involves time-sensitive information. In the application domain, certain classes of entities, such as countries, are considered non-varying entities. In these cases, no additional proxy representation is required. In ambiguous cases (i.e., a clause consisting solely of subject, predicate, and object variables), a clause is considered time-sensitive and treated appropriately. When an entity is proxied in this phase of query rewriting, a new proxy variable representing that entity is added to the variable bindings set for the query.

Care is taken when adding the proxy representation so as to not alter the underlying query logic. Expansion exactly matches the original structure of the query, respecting the original query block scoping for each entity and clause. For example, if an entity is only referred to in an `Optional` block of a query, the proxy expansion for that entity will only appear in the that `Optional` block. In this manner, the original query logic is left intact.

The temporal selection block is appended to the query in the topmost query block where the entity is proxied, respecting the original query logic as with proxy expansion. The temporal block's structure is dictated by the knowledge store's temporal index processor and incorporates the submitted time.

4 Example Scenario Query Results

To illustrate the impact of using our knowledge representation scheme and query rewriting service, let us briefly consider the example first presented in Section 2. Submitting the example query from Section 3 to the query rewriting service with three different time instants and then issuing those queries to the underlying knowledge store produces the results depicted in Table 2.

Note that, as mentioned in Section 3, the results of the rewritten query leave the original query bindings in-place and add proxy variables to capture the proxied state of an entity. This means that when considering the results of issuing the original query with the date-time of `2009-08-18T09:00:00` (the second row in the results table), the following interpretation is the correct one: There exists one `:Person` in the knowledge store with an SSN of `123-45-6789`. That person has two known names ("Robert Jones", "Bob Jones") and represents the merge

of two primitives that were considered, for the given time, to be references to the same real-world entity (represented by `:Proxy2`).

Table 2. Query results for the example scenario with different provided times

Provided Time	Query Results		
	?person_proxy	?person	?name
2009-08-17 12:00:00	`:Proxy1`	`:Person1`	"Robert Jones"
2009-08-18 09:00:00	`:Proxy2`	`:Person1`	"Robert Jones"
	`:Proxy2`	`:Person2`	"Bob Jones"
2009-08-18 09:40:23	`:Proxy4`	`:Person2`	"Bob Jones"

5 Conclusion

As evidenced by the scenario presented in Section 4, the system allows users to consider previous valid states of the knowledge store based on times of interest. This feature enables analysts to explore previous analyses in the context of what was known *then,* rather than what is known *now.*

Our system allows for evolution of knowledge while preserving all previous states of a knowledge store for subsequent review and investigation. Additionally, our knowledge representation scheme provides the additional benefit of non-destructive, reversible coreference resolution. These features are essential for conducting analysis in real-world, dynamically-evolving data environments.

References

1. Allen, J.F.: Towards a general theory of action and time. Artificial Intelligence 23(2), 123–154 (1984), http://www.sciencedirect.com/science/article/B6TYF-4811T47-4R/2/27f611303bc842936faa7f168fdcb9da
2. Kolas, D., Emmons, I., Dean, M.: Efficient Linked-List RDF Indexing in Parliament. In: Proceedings of the Fifth International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS2009). Lecture Notes in Computer Science, vol. 5823, pp. 17–32. Springer, Washington, DC (October 2009), http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-517/ssws09-paper2.pdf
3. McCarthy, J., Hayes, P.J.: Some Philosophical Problems from the Standpoint of Artificial Intelligence. In: Meltzer, B., Michie, D. (eds.) Machine Intelligence 4, pp. 463–502. Edinburgh University Press (1969), http://www-formal.stanford.edu/jmc/mcchay69.html, reprinted in McC90
4. Özsoyoglu, G., Snodgrass, R.T.: Temporal and Real-Time Databases: A Survey. IEEE Transactions on Knowledge and Data Engineering 7(4), 513–532 (August 1995), http://www.cs.arizona.edu/ rts/pubs/TKDEAug95.pdf
5. Smith, B., Grenon, P.: Basic Formal Ontology (BFO) (June 2010), http://www.ifomis.org/bfo

Introducing Ontological Realism for Semi-Supervised Detection and Annotation of Operationally Significant Activity in Surveillance Videos

Werner Ceusters[1], Jason Corso[2], Yun Fu[2],
Michalis Petropoulos[2], Venkat Krovi[3]

[1] Ontology Research Group, NYS Center of Excellence in Bioinformatics & Life Sciences, 701 Ellicott Street, Buffalo NY
[2] Department of Computer Science and Engineering, University at Buffalo
[3] Department of Mechanical and Aerospace Engineering, University at Buffalo
{ceusters, jcorso, yunfu, mpetropo, vkrovi}@buffalo.edu

Abstract. As part of DARPA's Mind's Eye program, a video-analysis software platform able to detect operationally significant activity in videos is being developed. The goal is to describe such activity semi-automatically in terms of verb phrases mapped to a realism-based ontology that can be used to infer and even predict further activities that are not directly visible. We describe how Region Connection Calculus and its derivative, Motion Class Calculus, can be used together to link the spatiotemporal changes that pixel-aggregates undergo in video-displays to the corresponding changes of the objects in reality that were recorded and to linguistic descriptions thereof. We discuss how Ontological Realism can be used as a safeguard to drawing such correspondences naively.

Keywords: ontological realism, video analysis, activity detection

1 Introduction

Automatic video-understanding is a relatively new field for which the research agenda has been set only fairly recently. Cetin identified in 2005 two grand challenges for video-analysis: the first was to develop applications that allow a natural high-level interaction with multimedia databases; the second was finding adequate algorithms for detecting and interpreting humans and human behavior in videos containing also audio and text information [1]. Early successes have focused on particular sub-problems, such as face detection [2].

State of the art systems are capable of detecting instances of objects – sometimes referred to as 'the nouns' of the scene – among few hundreds of object classes [3] and contests such as the PASCAL Challenge annually pit the world's best object detection methods on novel datasets [4]. Now, however, a more elusive problem presents itself: finding the 'verbs' of the scene. As Biederman stated nearly 30 years ago: specifying not only the elements in an image but also the manner in which they are interacting

and relating to one another is integral to full image understanding [5]. However, recognizing human actions, especially with a view to understanding their underlying motivation, has proved to be an extremely challenging task. This is because (1) behavior is the result of a complex combination of coordinated actions, (2) motion and behavior are described linguistically at a wide variety of spatiotemporal scales, and most importantly (3) the unambiguous extraction of intent from motion alone can never be achieved due to the significant dependence upon contextual knowledge.

Solving these problems, specifically in the context of surveillance, is the objective of DARPA's *Mind's Eye* program which seeks to embed in a *smart camera* sufficient visual intelligence to detect, interpolate and even predict activities in an area of observation and, as a specific requirement, to describe these activities in terms of 'verbs' (Table 1) [6].

As successful proposers to this program, our answer is ISTARE: a platform which will suitably represent articulated motion in a three-layer hierarchical dynamical graphical model consisting of (1) a lowest level of representation formed by points, lines and regions in their spatiotemporal context, (2) a mid-level capturing the spatio-temporal coherence inherent in the appearance, structure and motions of the atoms in the lower level, and (3) generalizations of the reusable mid-level parts into full objects and activities at the high-level (Fig.1). Part of that platform is an ontology which grounds the models with proper semantics thereby driving both learning and inference. A human-in-the-loop is the bridge between models and symbolic representations in case of ambiguities. But rather than requiring laborious annotation in such case, the human simply needs to answer yes/no questions generated by our methods.

In this communication, we describe our strategy to make the ISTARE approach in general, and the computational structures resulting from automated video analysis and annotation within the ISTARE platform specifically, compatible with ongoing research in the field. Using Motion Classes (MC) as an example, we demonstrate how Ontological Realism is an important building block in this endeavor and how it is able to tie the various pieces – reality, spatiotemporal models and linguistic descriptions – together.

Table 1. Verbs of interest for activity detection in the Mind's Eye video-analysis program

approach	catch	enter	follow	have	lift	put down	stop
arrive	chase	exchange	get	hit	move	raise	take
attach	close	exit	give	hold	open	receive	throw
bounce	collide	fall	go	kick	pass	replace	touch
bury	dig	flee	hand	jump	pick up	run	turn
carry	drop	fly	haul	leave	push	snatch	walk

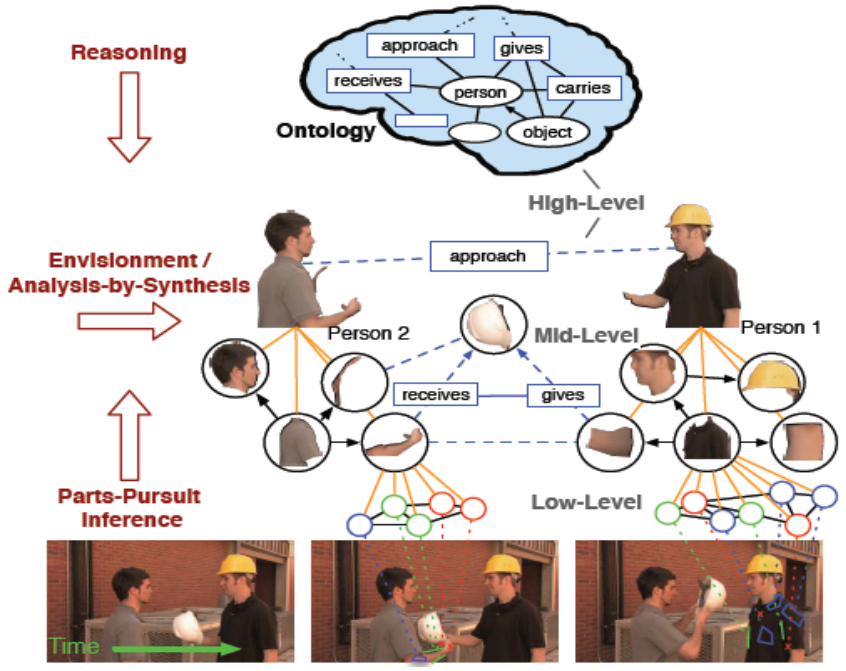


Fig. 1: Interaction of spatio-temporal models at three levels of granularity within the ISTARE platform.

2 Motion Classes

Several formalisms have been introduced to represent and reason with actions. The basic elements of situation calculus [7-8] are: (1) *actions* that can be performed in the world, (2) *fluents* that describe the state of the world, each fluent thus being the representation of some property, and (3) *situations*, a situation being '*a complete state of the universe at an instant of time*' [9], a position which is also maintained in fluent calculus [10]. Event calculus does without situations, and uses only actions and fluents, whereby the latter are functions – rather than predicates as is the case in situation calculus – which can be used in predicates such as *HoldsAt* to state at what time which fluents hold [11]. These approaches, unfortunately, don't take ontological commitment very serious or are based on representational artifacts which do not follow the principles of ontological realism [12].

The Motion Classes (MC) paradigm [13] which builds further on Region Connection Calculus (RCC) [14] to describe motions do not suffer from this. RCC describes how two regions are spatially located in relation to each other, thereby recognizing eight relations (Fig.2). Five of them are ontologically distinct: disconnected (DC), externally connected (EC), partially overlapping (PO), tangential proper part (TPP) and non-tangential proper part (NTPP). Three others are there for

notational purposes: equality (EQ – if we write 'EQ(x,y)', then there is in fact only one region denoted by two distinct symbols 'x' and 'y'), and TPPI and NTPPI as the inverses of TPP and NTPP.

The Motion Classes paradigm exploits what is called the 'conceptual neighborhood' of the RCC-relations which for each relation is defined as the set of possible relations that may hold at some point in time when another relation held earlier (Fig.2). A motion class is the set of transitions from one RCC configuration to another one that can be achieved by the same basic sort of motion (Fig.3). As an example, any transition from PO, TPP NTPP, EQ, TPPI and NTPPI to DC, EC or PO can be achieved through a LEAVE motion, i.e. a motion which separates the two regions from each other. Although there are 64 (8^2) distinct types of transitions which thus theoretically could be caused by 64 distinct types of motions, closer inspection reveals that there are only nine distinct motion types (Fig.3). Five more distinct classes can be defined through pair-wise combination of the nine basic motions: HIT-SPLIT, PERIPHERAL-REACH, PERIPHERAL-LEAVE, REACH-LEAVE, and LEAVE-REACH. The 76 (9^2-5) other combinations do not lead to a distinct sort of motion; HIT followed by REACH, for instance, was already REACH from the very beginning.

In the same way as RCC calculus uses tables to list the possible configurations for region pair (x,z) when the RCC-relations for the pairs (x,y) and (y,z) are known, so provides MC tables for what motion classes are possible for the pair (x,z) when the motion classes for (x,y) and (y,z) are known [13]. MC, in addition to being a representational framework for motion, can also be used as the semantic underpinning for motion verbs. Almost all verbs from Table 1 can be analyzed in terms of a motion class: 'leave' and 'give' involve LEAVE, 'hit' and 'collide' involve HIT, 'bounce' involves HIT-SPLIT, 'approach' involves REACH, and so forth. The feasibility of this approach has already been determined although some further representational frameworks for spatiotemporal reference and direction are required [15]. But, as we will discuss in section 4, an adequate ontological analysis as applied in related contributions to geographic information science [16], is required to determine precisely what sort of involvement is the case.

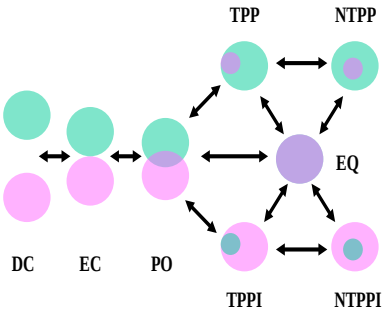


Fig.2: Relationships and transitions involving spatial regions in Region Connection Calculus.

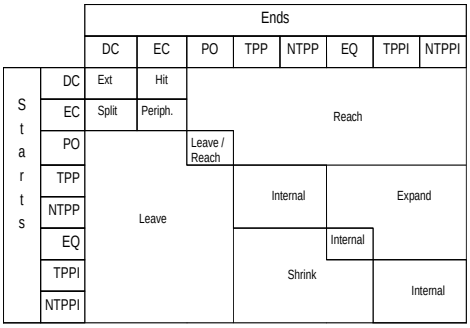


Fig.3: 9 basic motion classes representing the simplest type of change that two regions possibly underwent relative to their start and end configuration as expressed in RCC8.

3. Ontological Realism

Ontological realism is a paradigm which rests on a view of the world as consisting of entities of at least two sorts, called 'particulars' and 'universals' respectively [12, 17]. *Particulars*, according to this doctrine, are concrete individual entities that exist in space and time and that exist only once, examples being the persons and helmets depicted in Fig.1. Persons and helmets are *continuants*: whenever they exist, they exist in total. Also the motion in which the white helmet participated ('*participate'* is here a technical term which expresses in the least informative or committing way the relationship between a continuant and the change it undergoes [18]) while being given to the person depicted on the left is a particular. Particulars such as motions are *occurrents*, i.e. entities that at every point in time that they exist, exist only partially.

The word '*depicted'* in the sentence above is not used arbitrarily, because the three bottom images in Fig.1 are themselves three distinct particulars and so are the parts of these pictures which depict the persons. But whereas the persons themselves are not about anything, the corresponding pictures are about these persons. Therefore, the persons are so-called *L1-entities* (first-order entities) while the pictures are *L3-entities*, i.e. communicable representations [19]. It is this communicability that distinguishes L3-entities from cognitive representations (*L2-entities*) such as beliefs, for example the belief sustained by an intelligence analyst that the person on the left in each of these images is the same person, or the belief that this person is John Smith. The analyst can of course express his belief in an annotation to the pictures, that annotation then being an L3-entity and thus clearly distinct from the belief itself: the belief is in the analyst's head, the annotation is in the report.

Universals, in contrast to particulars, are repeatable. This means that they that can exist in indefinitely many instances – thus the persons depicted in Fig.1 were instances of the universal *HUMAN BEING* at the time the pictures (each of the latter being instances of the universal *PICTURE*) were taken – and they are the sorts of things that can be represented by means of general terms used in the formulation of theories, for instance that pictures shot by good cameras contain regions of which the colors correspond with the colors exhibited by the entities in reality to which those regions correspond.

Ontological realism is embodied in two artifacts which roughly correspond with the universal/particular distinction. *Basic Formal Ontology* (BFO) [20] represents the universals which are practically necessary for successful domain and application ontology construction and ensures (1) that there is an unbroken path from every type in the ontology to the ontology's root, and (2) that definitions for all terms in the ontology can be coherently formulated. *Referent Tracking* (RT) provides a set of templates to express formally *portions of reality* (PORs), i.e. how particulars relate to each other, what universals represented in BFO (or ontologies developed there from such as for instance UCORE-SL [21]) they instantiate, and what terms from other terminological systems are used to further describe them [22].

4. Video, spatiotemporal semantics and ontological realism

How do videos of PORs and descriptions about these PORs on the basis of what is depicted in a video, relate to reality under the view of ontological realism?

Digital images taken from PORs contain pixels most of which combine into curves and regions which each have their own shape and texture, all of these entities being continuants. In the ideal case, regions in the image depict (roughly) the characteristics of the surface of the material entities visible to the camera which are all continuants too. Digital video files of PORs are continuants which when processed by display technology lead to the generation of occurrents of which the curves and regions, as well as their shapes and textures, are the only participants. These occurrents are the coming into existence, disappearance, or change in location, shape, size and/or texture of curves and regions. In the ideal case, with an immobile camera and without zooming having been applied, the occurrents visible on the screen correspond to occurrents in which the material entities that are depicted participate. But whereas the on-screen (L3) entities are instances of a very restricted number of universals, there are many more universals of which the corresponding L1-entities are instances. Furthermore, although each particular on-screen entity corresponds (roughly) to exactly one L1-entity, distinct on-screen entities in distinct images or videos may correspond to distinct L1-entities despite being of exactly similar shape, size and texture. Video-analysis can under this view thus be seen as an effort to identify (1) the on-screen regions and their changes which correspond to L1-particulars, (2) the universals instantiated by these L1-particulars, and (3) the identity of these particulars, either absolute (e.g. establishing that John Smith is depicted in the video) or relative (e.g. the person leaving the building is the same as the one that earlier entered the building).

Video-annotation under the Mind's Eye program requires the use of certain descriptive verbs (Table 1) which brings in additional complexity involving L2-entities. Not only is there a wide variability in the way motion classes are linguistically expressed [15], it has also been shown that the cognitive salience of topological relations is not equal for all topologically defined ending relations [23]. Various pitfalls need thus to be avoided. as demonstrated by a verb such as 'to approach'. One pitfall is leaving it open whether a descriptive verb is used to describe an on-screen entity or a first-order entity: although one on-screen entity might indeed be described as approaching another one, it might be such that the corresponding entities in reality are moving away from each other, the on-screen approach being the result of the reduction from 3D to 2D when reality is viewed at through the lens of a camera. Another pitfall is that some motion verbs behave grammatically as action verbs when used in the description of a scene, while in reality the process that is described as such, although ontologically being an instance of motion, is not at all an instance of action: the canoe floating down the river towards the lake is indeed approaching the lake, but without any action going on. Yet another pitfall is that two entities might be described as being involved in an approach although the shortest distance between them increases: think of two cars driving towards each other on a curved road around some mountain. It might be tempting to say that in this case there are two motions going on, one of approaching and one of moving away, but that is of course ontological nonsense. And as a last example, but for sure not the last pitfall,

many motion verbs do not *denote* motions at all, but rather certain configurations of entities in which some sort of motion is involved. 'To approach' is in this case. Imagine a satellite orbiting around Earth for years and that at some point in time a second satellite is launched in an orbit which is such that during some part of their motions the two satellites can be said to approach each other while during other parts they can be said to move away from each other. It seems obvious that the process in which the first satellite participated for all these years does suddenly not become a different process because of some event that does not have any effect on its motion. Yet, the descriptions are valid at the respective times.

Ongoing efforts

Automatically extracting from a video regions that correspond to concrete objects and parts of objects, and then identifying what these objects exactly are, is a challenging problem. Although progress toward object boundary segmentation at the low-level continues to be made, all sufficiently successful approaches are either limited to specific object classes or have not been applied to videos. To overcome these challenges, ISTARE works currently with a hierarchy of pixel aggregates which is induced directly from the pixel data and hence, does not impose an artificial structure. At the low-level, direct pixel intensities are used to decide whether or not to join into an aggregate, and thus do suffer from the above noted limitations. But, as the algorithm moves up the hierarchies, more informative features, such as texture and shape, are used to describe the aggregates and hence assist in deciding which ones should be joined. The aggregation occurs directly on spatiotemporal pixel cubes, defined over short segments of the video (e.g., 2 seconds) [24]. The goal is now to improve the recognition at this level by using information provided by an ontology developed along the lines just sketched.

Acknowledgments. The work described was funded in part by DARPA's Mind's Eye Program and the Army Research Lab under grant W911NF-10-2-0062.

References

1. Cetin, E., *Interim report on progress with respect to partial solutions, gaps in know-how and intermediate challenges of the NoE MUSCLE*. 2005.
2. Viola, P. and M.J. Jones, *Robust real-time face detection*. International Journal on Computer Vision, 2004. **57**(23): p. 137-154.
3. Fei-Fei, L., R. Fergus, and P. Perona., *Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories*, in *IEEE Conference on Computer Vision and Pattern Recognition Workshop on Generative-Model Based Vision*. 2004, IEEE.
4. Everingham, M., et al., *The Visual Object Classes (VOC) Challenge*. International Journal of Computer Vision, 2010. **88**(2): p. 303-338.
5. Biederman, I., *On the Semantics of a Glance at a Scene*, in *Perceptual Organization*, M. Kubovy and K.R. Pomerantz, Editors. 1981, Lawrence Erlbaum Publisher. p. 213-263.

6. Defense Advanced Research Projects Agency. *Mind's Eye Broad Agency Announcement*. 2010 [cited 2010 August 10]; Available from: http://www.darpa.mil/tcto/solicitations/BAA-10-53.html.
7. McCarthy, J., *Situations, actions and causal laws*. 1963, Stanford University Artificial Intelligence Laboratory: Stanford, CA.
8. Reiter, R., *The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression*, in *Artificial intelligence and mathematical theory of computation: papers in honour of John McCarthy*, V. Lifshitz, Editor. 1991, Academic Press Professional, Inc: San Diego, CA, USA. p. 359-380.
9. McCarthy, J. and P.J. Hayes, *Some philosophical problems from the standpoint of artificial intelligence.* Machine Intelligence, 1969. **4**: p. 463-502.
10. Thielscher, M., *Introduction to the Fluent Calculus.* Electronic Transactions on Artificial Intelligence, 1998. **2**(3-4): p. 179-192.
11. Kowalski, R., *Database updates in the event calculus.* Journal of Logic Programming, 1992. **12**(1-2): p. 121-46.
12. Smith, B. and W. Ceusters, *Ontological Realism as a Methodology for Coordinated Evolution of Scientific Ontologies.* Journal of Applied Ontology, 2010 (in press).
13. Ibrahim, Z. and A. Tawfik, *An Abstract Theory and Ontology of Motion Based on the Regions Connection Calculus*, in *Abstraction, Reformulation, and Approximation*, I. Miguel and W. Ruml, Editors. 2007, Springer Berlin / Heidelberg. p. 230-242.
14. Randell, D.A., Z. Cui, and A.G. Cohn, *A spatial logic based on regions and connection*, in *Proceedings of the Third International Conference on the Principles of Knowledge Representation and Reasoning*, B. Nebel, W. Swartout, and C. Rich, Editors. 1992, Morgan Kaufmann: Los Altos, CA. p. 165-176.
15. Pustejovsky, J. and J.L. Moszkowicz, *Integrating Motion Predicate Classes with Spatial and Temporal Annotations*, in *Coling 2008: Companion volume: Posters*. 2008, Coling 2008 Organizing Committee: Manchester. p. 95--98.
16. Worboys, M.F. and K. Hornsby, *From objects to events: GEM, the geospatial event model*, in *GIScience : Proceedings of the Third International Conference on GIScience*, M.J. Egenhofer, C. Freksa, and H. Miller, Editors. 2004, Springer Verlag: Berlin. p. 327-44.
17. Smith, B. and W. Ceusters, *Towards Industrial-Strength Philosophy; How Analytical Ontology Can Help Medical Informatics.* Interdisciplinary Science Reviews, 2003. **28**(2): p. 106-111.
18. Smith, B., et al., *Relations in biomedical ontologies.* Genome Biology, 2005. **6**(5): p. R46.
19. Smith, B., et al., *Towards a Reference Terminology for Ontology Research and Development in the Biomedical Domain*, in *KR-MED 2006, Biomedical Ontology in Action.* 2006: Baltimore MD, USA
20. Grenon, P. and B. Smith, *SNAP and SPAN: Towards dynamic spatial ontology.* Spatial Cognition and Computation, 2004. **4**(1): p. 69-103.
21. Smith, B., L. Vizenor, and J. Schoening, *Universal Core Semantic Layer*, in *OIC-2009: Ontologies for the Intelligence Community*, P. Costa, K. Laskey, and L. Obrst, Editors. 2009, CEUR: Fairfax, VA.
22. Ceusters, W. and S. Manzoor, *How to track Absolutely Everyting?*, in *Ontologies and Semantic Technologies for the Intelligence Community. Frontiers in Artificial Intelligence and Applications.*, L. Obrst, T. Janssen, and W. Ceusters, Editors. 2010, IOS Press: Amsterdam. p. 13-36.
23. Klippel, A. and R. Li, *The endpoint hypothesis: A topological-cognitive assessment of geographic scale movement patterns.* Conference on Spatial Information Theory (COSIT 2009), 2009: p. 177-194.
24. Corso, J.J., et al., *Efficient multilevel brain tumor segmentation with integrated bayesian model classification.* IEEE Transactions on Medical Imaging, 2008. **27**(5): p. 629-640.

Ontological Constructs to Create Money Laundering Schemes

Murad Mehmet and Duminda Wijesekera
George Mason University Fairfax, VA 22030
mmehmet@gmu.edu, dwijesek@gmu.edu

Abstract. There is an increasing tendency in the money laundering sector to utilize electronic commerce and web services. Misuse of web services and electronic money transfers occurs at many points in complex trading schemes. We provide ontological components that can be combined to construct some of these money laundering schemes. These constructs can be helpful for investigators, in order to decompose suspected financial schemes and recognize financial misuses.

Keywords: money laundering, money laundering ontology.

1 Introduction

Money Laundering Schemes (MLS) have evolved in order to take advantage of internet based financial transactions and web services. To date, regulations alone have not been able to deter such schemes, as seen in recent examples of long running money laundering schemes [1], [2], [3], [4]. Digital currencies (E-Money) are particularly suitable for money laundering schemes because of their global usability, anonymity, ease of use, and instantaneous transferability. It is becoming increasingly difficult to differentiate between legitimate and fraudulent transactions because of their complexity and evolving nature, as described in recent publications [3], [4], [9].

In order to decompose this complexity we provide some basic ontological constructs that can be used to create known money laundering schemes. These basic ontological constructs can be integrated with financial transaction specification languages to provide further forensic analysis, particularly with XBRL, the de-facto standard for reporting in the financial industry, in order to recognize financial misuses.

The rest of the paper is organized as follows: Section 2 discusses the well known money laundering schemes. Section 3 defines the proposed money laundering ontological constructs. Section 4 presents an example of constructing a money laundering scheme using the proposed ontological constructs. Section 5 presents a discussion on related work in the area of money laundering ontologies. Finally, section 6 presents the conclusion.

2 Known Money Laundering Schemes

In order to identify the basic components of existing money laundering schemes, we list some well known money laundering schemes as follows:

1. Structured Transfer Scheme: This method involves splitting a transfer of funds into multiple fund transfers involving smaller amounts that are below the threshold of suspicion.
2. Alternative Remittance Systems Scheme: In this method, all transactions are done in cash involving parties (two or more) that calculate the difference of their balances, and make quick payments in their own countries without involving any electronic wire transfer.
3. Loan Back Scheme: In this method, a shell company (a fictitious company created merely to transfer money without raising suspicion) transfers funds allocated as credit from the money launderer in the form of a loan. The loan is then repaid with laundered money, thereby legitimizing the laundered money.
4. Low Invoicing Scheme: In this method, the seller lowers the invoice to the buyer as payment for an illegal commodity (such as drugs or weapons). The buyer then resells the product for a high profit.
5. High Invoicing Scheme: In this method, high prices for goods are paid by contractors resulting in high profits (laundered money) for the seller. It is characterized by fabricated deliveries of products, transactions carried out by shell companies in offshore territories, and use of electronic payments by anonymous persons.
6. Anonymous Account Holder Services: In this method, accounts are created by E-Money servers for customers who wish to be anonymous during the use of E-Money transactions. These are attractive to money launderers due to the ease and secrecy of fund transfers among the accounts, as well as the accessibility to fund withdrawals at any regular banking locations.

3 Components of Money Laundering Schemes

The four basic entities that we use to construct a money laundering scheme are people, organization, portfolio, and messages. The “people” represents the individuals who participate in a business transaction. This entity can be business related, non-business related, or a money launderer. The “organization” represents any institution or firm that engages in financial operation or business trading. The “portfolio” represents any asset of a person or an organization in a financial institution. “Messages” represents any form of communication exchanged between people and organizations.

We also use three auxiliary entities: communication medium, invoice and identification documents to represent schemes. The “communication medium” represents any environment that allows the delivery of messages. The “invoice”

represents the demand for payment issued in trading schemes. "Identification documents" are used to identify the "people".

There are many relationships amongst the entities. Therefore, we formally define these entities and relationships using the Web Ontology Language (OWL).

3.1 The Ontology of Money Laundering Schemes

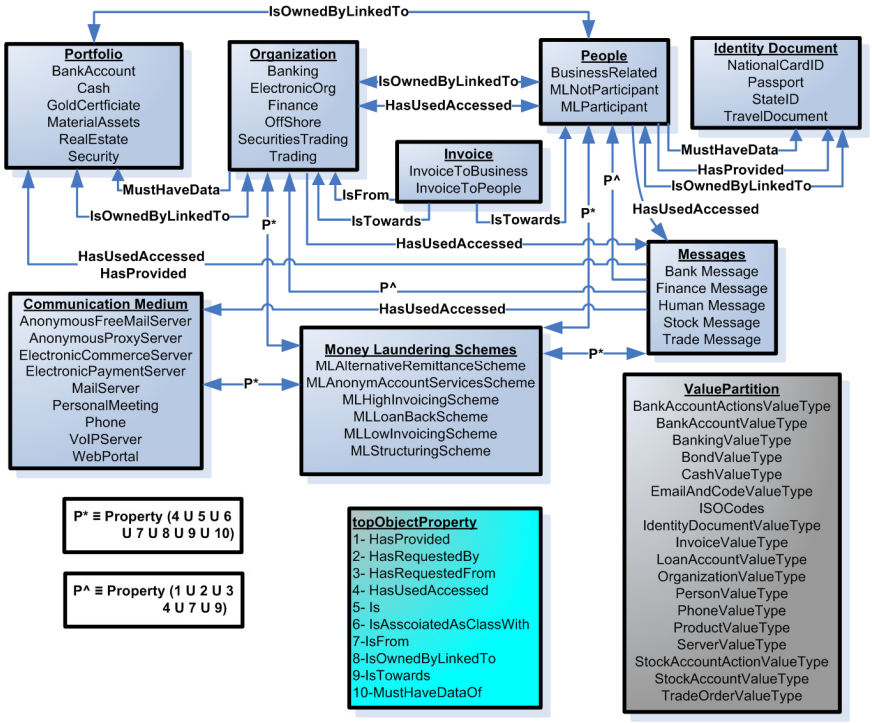


Fig. 1. The ontology class diagram

The ontology class diagram represents the components used in money laundering schemes as described in previous section: people, organization, portfolios, messages, communication medium, invoice and identification documents. We list and describe the entities in the OWL ontology shown in Figure 1 above as follows:

1. People: Represents individuals who participate in a business transaction. It consists of the subclasses "business related", "ML participant", and "ML not participant". This entity is associated with entities: "organization", "portfolio", "messages", "identification document", and "money laundering schemes". For instance, money launderer "people" need "identification documents", to send "messages" to withdraw "portfolio" cash from an "organization" bank, as part of a structuring "money laundering scheme".

2. Portfolio: Represents financial assets and products, it has many subclasses such as "cash", "security", and "bank account". This entity is associated with "organization", "people" and "messages". For instance, "people" own accounts linked to a bank "organization", they access them via bank transaction "messages".
3. Organization: Represents any business engaged in trading or financial transactions, it has many subclasses such as "banking", "securities trading", and "electronic organization". This entity is associated with entities: "people", "portfolio", "messages", "invoice", and "money laundering schemes". For instance, a security trading company sends invest "messages", or issues an "invoice" from the security account to the account owner.
4. Messages: Represents all messages exchanged in the domain between "people" and "organization". All the activities in the domain are performed via messages, such as "bank messages", "trade messages", and "human messages". This entity is associated with entities: "people", "organization", "portfolio", "communication medium" and "money laundering schemes". For instance, to withdraw funds the money launderer "people" send the withdraw "message" to the bank "organization", and thereby the withdraw "message" accesses the "portfolio" bank account, as part of the structuring "money laundering scheme" using the phone "communication medium".
5. Communication medium: Represents all methods of standard and encrypted communication. It has many subclasses such as "anonymous proxy server", "electronic payment server", and "mail server". This entity is associated with entities "message" and "money laundering schemes". For instance, the deposit uses the "electronic payment server" as part of the "money laundering schemes".
6. Identification document: Represents all documents that can be provided by the person for identification purposes. It has many subclasses such as "national card ID" and "passport". This entity is only associated with the entity "people". For instance, money launderer "people" must have an "identification document" passport.
7. Invoices: Represents trading statements. It consists of the subclasses "invoice to business" and "invoice to people". This entity is associated with entities "organization" and "people". For instance, an "organization" issues an "invoice" to "people".
8. Money laundering schemes: Represents the various money laundering techniques, it has many subclasses such as "low invoicing scheme" and "structuring scheme". The finance industry is very dynamic, as the money laundering techniques continue to evolve they will be added to our ontology. This entity is associated with entities: "people", "organization", "message", and "communication medium". For instance, money launderer "people" send transfer "message" to bank "organization", as part of the high invoicing "money laundering schemes".

We list and describe the object properties in the OWL ontology as follows:

1. HasProvided: For one entity to provide information to another entity. For instance, a person provides his or her bank account number to an organization.

2. HasRequestedBy: An entity makes a request to another entity. For instance, an EFT is requested by an account holder from a bank.
3. HasRequestedFrom: An entity receives a request from another. For instance, an EFT requested from a bank by a person.
4. HasUsedAccessed: An entity uses or accesses another entity.
5. Is: To associate an entity within the MLS with their specific entity. For instance, the entity "EMSS Launderer" is a "MLSParticipant".
6. IsAssociatedAsClassWith: To associate or link an entity "Value Type" with its super class.
7. IsFrom: To associate the source entity of messages that is not in the form of a request. For instance, an electronic fund transfer is from a person.
8. IsOwnedByLinkedTo: An entity that is owned by or linked to another.
9. IsTowards: To associate the target entity of messages that is not in the form of a request. For instance, an electronic fund transfer is towards a shell company.
10. MustHaveDataOf: An entity has data of another. For instance, a bank must have data of the account holder person.

4 Example Construction of Money Laundering Scheme

In this section we create the anonymous account holder services scheme, using the constructs from our OWL ontology. According to our OWL definition, messages are linked to one or more entities. For instance, opening an account is a relation linked to the requester entity and the requested entity, the request message is sent by a person to a bank. Another example can be the relation in electronic fund transfer (EFT), where there is a receiver entity and a sender entity. Owning an account, however, is linked to only one entity.

We list the message sequence of the example scheme in Table 1.

Table 1. Choreographies of Anonymous Account Holder Services Scheme

Step	Entity	Message (Linked Entity)
1st	AnonySession	HasRequestedFrom(ProxyServer), HasRequestedBy(MLaunderer)
2nd	AnonySession	IsFrom(ProxyServer), IsTowards(MLaunderer)
3rd	EMAccount-1	HasRequestedFrom(EMoneyServer), HasRequestedBy(MLaunderer)
4th	EMAccount-1	IsFrom(EMoneyServer), IsTowards(MLaunderer)
5th	MLaunderer	HasProvided(ShellComp)
6th	ShellComp	MustHaveDataOf(EMAccount-1)
7th	MLaunderer	HasUsedAccessed (DepositCash)
8th	DepositCash	IsFrom(MLaunderer), IsTowards(ShellComp)
9th	EMAccount-2	HasRequestedFrom(EMoneyServer), HasRequestedBy(ShellComp)
10th	EMAccount-2	IsFrom(EMoneyServer), IsTowards(ShellComp)
11th	E-Deposit	HasRequestedFrom(EMExchange), HasRequestedBy(ShellComp)
12th	EMExchange	HasUsedAccessed(EMAccount-2)
13th	EFT	IsFrom(EMAccount-2), IsTowards(EMAccount-1)

14th	EFT	IsFrom(ShellComp), IsTowards(MLaunderer)
15th	Withdraw	HasRequestedFrom(EMExchange), HasRequestedBy(MLaunderer)
16th	Withdraw	HasUsedAccessed(EMoneyServer)
17th	Withdraw	IsFrom (EMAccount-1), IsTowards(MLaunderer)

We briefly describe the choreographies of Table 1 as follows:

Steps 1 and 2 are the request for an anonymous session by the money launderer and the opening of the session by the proxy server. Steps 3 and 4 are the request for an electronic currency account by the money launderer and the opening of the account by the electronic payment server. In steps 5 and 6 the money launderer passes the account information to the shell company. In steps 7 and 8 the money launderer transfers cash funds to the shell company. Steps 9 and 10 are the request for an electronic currency account by the shell company and the opening of the account by the electronic payment server. Steps 11 and 12 represent the cash deposit of the shell company to the electronic currency exchange and provision of account information. Steps 13 and 14 are the transfer of funds from the shell company to the money launderer using electronic currency accounts. Steps 15, 16, and 17 represent the withdrawal of funds from the electronic currency account of the money launderer, using the electronic currency exchange office.

Figure 2 represents the sequence diagram of the choreographies, using the relation and constructs from the OWL ontology. Figure 3 depicts the objects properties used in the ontology, linking the entities of the choreographies of anonymous account holder services scheme.

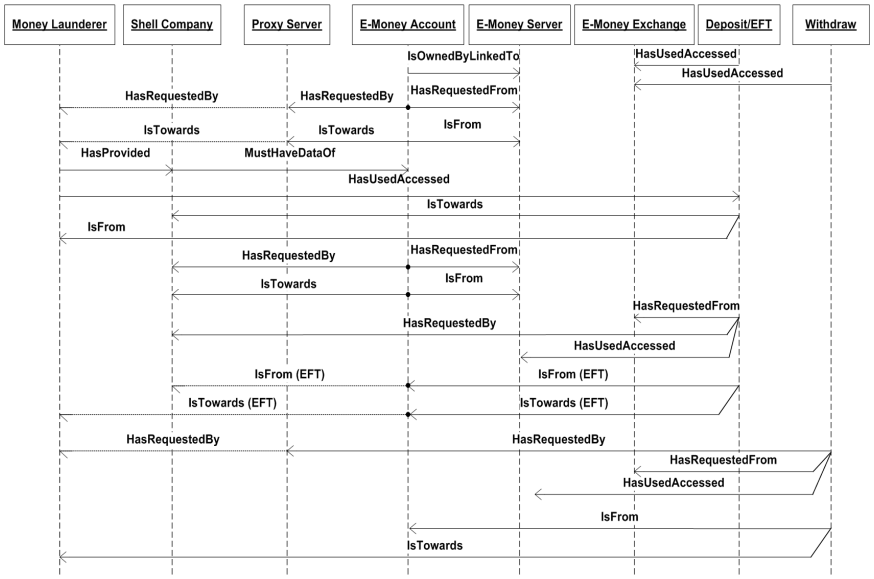


Fig. 2. The equence diagram of the anonymous account holder services scheme

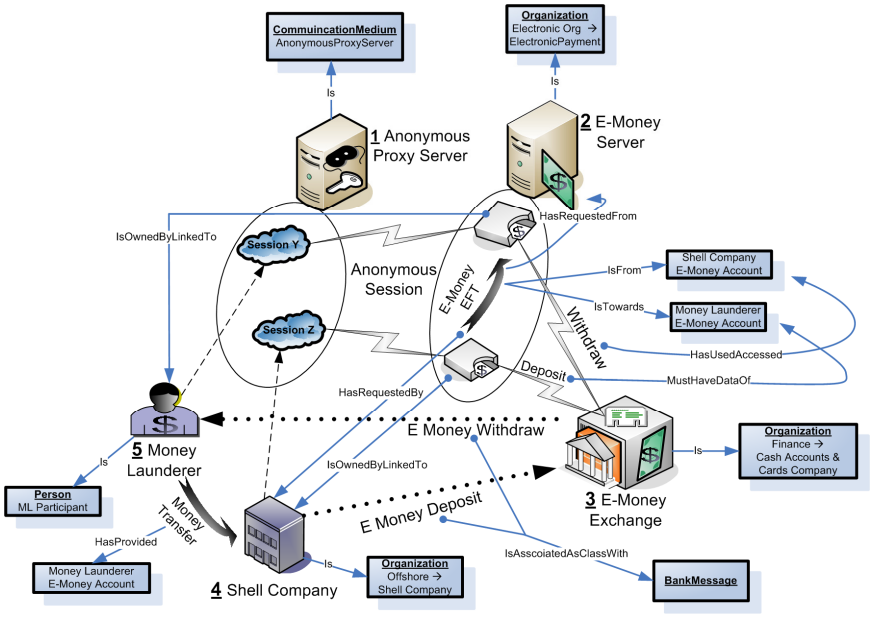


Fig. 3. The objects represented in the anonymous account holder services scheme

5 Related Work

International organizations such as The Financial Action Task Force on Money Laundering and National Drug Intelligence Center publish annual reports and statistics of money laundering trends, including ongoing investigation of cyber laundering cases [1], [2], [3], [4]. FF-POIROT [5], [6], [7], [8] is a project which builds a detailed ontology of European law on the preventive practices of financial fraud. The project is focused on sales tax fraud and online investment solicitation, and it does not go into details of money laundering ontologies and schemes. Woda [9] extensively describes money laundering techniques, but does not include any formal specification or ontology definition of the MLS. Vanderlinden [12] produced a comprehensive OWL ontology for financial systems, and covers legitimate transactions. The emphasis of the work done is to produce the OWL, and no detail is provided about the formal definition and methodological background.

Several publications study deficiencies of the languages used in the financial industry, with a particular focus on the taxonomy and the specification of the reporting languages. None of these studies cover the MLS with the exception of Viveo [20] and SEPBLAC [21].

As part of their consulting work for large global financial enterprises, Viveo released their product “QUALIFY-IT- XBRL Reporting” to provide bankers with uniform message content (e.g. Fraud detection, Risk control, Money laundering)

before anyone else can get it. The Viveo [20] product is tailored to the retail banking industry and heavily depends on XBRL [13], and thus lacks the capability to be used in web services transaction languages such as IFX [22]. The taxonomy project of SEPBLAC [21] entitled "Telematic Reporting Project" automates the reporting process of suspicious transactions, improve efficiency with fewer tasks and errors, and ensure scalability. Chen et al. [13] assess different taxonomies used for financial reporting in different countries, based on data samples selected from the Shanghai Stock Exchange. They explore if the current XBRL can apply to real life scenarios, and conclude the need to improve XBRL. Nicola et al. [14] developed an application-oriented, domain-specific benchmark "Transaction Processing over XML", which simulates multi-user financial workloads with data based on the FIXML standard. Carrillo et al. [15], [16] propose creating middleware to reduce the incompatibility from multiple implementations of XBRL in an enterprise. This is based on their developing an XBRL taxonomy for public institutions in Colombia.

Several efforts are underway in developing taxonomies for financial and investment organizations. Progress is being made on preparing taxonomy for the financial industry and investment organizations. Lara et al. [17] introduce a generic translation process of XBRL taxonomies of investment funds into OWL ontologies. They suggest that extensions to OWL are required to fulfill all the requirements of financial information reporting. An improved XBRL can be achieved by adding formal semantics. Castells et al. [18] developed an ontology-based platform that provides the integration of contents and semantics in a knowledge base that provides a conceptual view of low-level contents and semantic search facilities. Dui et al. [19] demonstrate that configuration management for XML languages is more complicated than traditional software engineering artifacts, they propose to evaluate XML by using different versions of the Financial Products Markup Language (FpML). They conclude that designers of FpML, and of many other complex XML languages, may need to make changes to the language while retaining overall compatibility. None of these works mentioned above analyze the semantics of money laundering, nor propose a model that can used to detect the schemes within the available financial reporting languages such as IFX [22], a language the financial industry heavily depends upon for web-based transaction and business-to-business banking.

We have used Methontology [10] to develop this ontology because Protégé [11] uses it.

6 Conclusions

In this paper we describe a preliminary OWL ontology to build money laundering schemes. Our ontology provides components that can be used to construct MLS. Our work creating money laundering ontologies is aimed at providing formal semantics for financial transaction data, and facilitating detection of illegal financial schemes. Currently, we are working on developing algorithms to detect each of the schemes from a sequence of financial transaction records, where the objective is to capture and identify the transactions that match constructs from our OWL ontology.

References

1. The Financial Action Task Force on Money Laundering: Annual Review of Non-Cooperative Countries or Territories. http://www.fatfgafi.org/dataoecd/3/52/33922473.pdf (2004)
2. The Financial Action Task Force on Money Laundering: Financial Action Task Force Annual Report 2008-2009. http://www.fatf-gafi.org/dataoecd/11/58/43384540.pdf (2009)
3. The Financial Action Task Force on Money Laundering: Money Laundering & Terrorist Financing Vulnerabilities of Commercial Websites and Internet Payment Systems. http://www.fatf-gafi.org/dataoecd/57/21/40997818.pdf (2008)
4. National Drug Intelligence Center: Money Laundering in Digital Currencies. http://www.justice.gov/ndic/pubs28/28675/index.htm (2008)
5. Zhao G., Kingston J., Kerremans K., Coppens F., Verlinden R.: Engineering an Ontology of Financial Securities Fraud. In: Workshop of Regulatory Ontology (2004)
6. Kerremans K., Tang Y., Temmerman R., Zhao G.: Towards Ontology-based E-mail Fraud Detection. In: 12th Portuguese Conference on Artificial Intelligence (2005)
7. Project Financial Fraud Prevention Oriented Information Resources Using Ontology Technology, http://www.ffpoirot.org (2010)
8. Leary R.M., Vandenberghe W., Zeleznikow J.: Towards a Financial Fraud Ontology: A Legal Modelling Approach. In : Workshop on Legal Ontologies, ICAIL (2003)
9. Woda, K.: Money Laundering Techniques with Electronic Payment Systems. In: Information & Security International Journal, vol .18, pp. 27--47 (2006)
10. Lopez M.F., Perez, A.G. : METHONTOLOGY: From Ontological Art Towards Ontological Engineering. In: Symposium on Ontological Engineering of AAAI, pp. 33--40 (1997)
11. Horridge M., Jupp, S., Moulton G.: A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools. http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial (2009)
12. Vanderlinden, E.: Finance Ontology and Semantic Technologies. http://www.fadyart.com/financeV4.owl (2010)
13. Chen, H.: Application and Neediness of Extensible Business Reporting Language. In: International Forum on Information Technology and Applications (2009)
14. Nicola, M. Kogan, I.: An XML transaction processing benchmark. In: 2007 ACM SIGMOD International Conference on Management of Data (2007)
15. Carrillo, E., Chaparro F., Santoyo, J.: XBRL and Financial Information Standards: a Case Success: University – Enterprise. In: Euro American Conference on Telematics and Information Systems (2008)
16. Carrillo E.: XBRL: From Common Financial Vocabularies to Intelligent Decision Making. In: Euro-American Conference on Telematics and Information Systems (2007)
17. Lara R., Cantador I., Castells P.: XBRL Taxonomies and OWL Ontologies for Investment Funds. In: 1st International Workshop on Ontologizing Industrial Standards (2006)
18. Castells, P., Foncillas B., Lara R., Rico M., Alonso, J.L. :Semantic Web Technologies for Economic and Financial Information Management. In: ESWS, pp. 473--487 (2004)
19. Dui D., Emmerich W.: Compatibility of XML Language Versions. In: Lecture Notes in Computer Science, pp. 148--162 (2003)
20. Viveo Systems: QUALIFY-IT- XBRL Reporting. In: XBRL-CEBS Workshop (2005)
21. SEPBLAC: Anti-Money-Laundering XBRL Taxonomy Project. In: 11th XBRL Conf (2005)
22. IFX Forum: Interactive Financial Exchange Message Specification 1.7.0. http://www.ifxforum.org/standards/standard (2005)

Enabling Rich Discovery of Web Services by Projecting Weak Semantics from Structural Specifications

Leo Obrst, Dru McCandless, Michael Bankston

The MITRE Corporation
{lobrst, mccandless, mbankston}@mitre.org

Abstract. Although we would prefer using defined ontologies that express the domains and specifications of web services, and thus more easily discover and compose these, we know that in the mainstream world represented by the US Department of Defense we will not have those ontologies available soon. In the meantime we have to ensure a transition from structural to semantic methods, including web service discovery methods. In this paper, we are proposing a different approach for dynamic web service discovery that takes advantage of the structure inherent in web services that are defined by WSDL documents. Since the structure is usually based on XML Schema, there is enough information present in these documents to develop a broadly applicable approach. Furthermore, if a consistent and detailed naming convention of schema artifacts is followed, then discovery can be made more precise. This paper describes our approach for projecting weak semantics from structural information for discovery of web services.

1 Introduction*

The use of web services has grown steadily over the past few years due to their ease of use and modularity for providing information in a standard way (including information retrieval, such as in a digital library). Web services have demonstrated their value for solving information needs that are part of regular, foreseen tasks. Thanks to standards such as the Web Service Description Language (WSDL) [8] and the Business Process Execution Language (BPEL) [9], and approaches such as Service Oriented Architecture (SOA), the way web services are defined and presented has also become more universal. However, many information integration tasks are unforeseen at the time the services are constructed, and are therefore difficult to perform "on the fly". Users lack the tools to search for what they want (i.e., the services that provide the specific information they

desire) and the tools to quickly and effectively examine a potential web service to determine if it meets their needs. The ability to quickly discover and chain web services to accomplish some dynamic information need is still a ways off, and will require additional technology before it is fully realized.

Discovery in this context is the ability to locate and understand a web service that is defined by the WSDL standard. This is usually a preliminary step before the service is accessed and used. Web service discovery is a long recognized problem, and several approaches have developed as attempts to solve it. The current standard for web service discovery is Universal Description, Discovery, and Integration (UDDI) [1], which uses tModels to describe a service's content. Although useful for locating services of which the searcher is already aware, it is decidedly less useful for discovering previously unknown services.

To alleviate this, much work in the area of Semantic Web services has been done. If a searcher were able to discover web services using the same tools and techniques for performing semantic queries, then better results could be obtained. Several efforts are showing promise in this area, such as FUSION [2, 3], WSMX [4], Service-Finder [5], and EASY [6-7]. However, the premise of many Semantic Web approaches is that the source information is first organized into an underlying ontology, and then the service definitions are derived from that ontology. Unfortunately, if the organization creating the web service doesn't use this approach (i.e., have an underlying ontology) then the search and discovery benefits cannot be realized. In the current technology environment (particularly within the US Department of Defense) while web services abound, the use of ontologies is being only slowly adopted, so relying on their use for service discovery won't work. The reasons for this are varied: lack of requirements, mismatch in skill sets between ontology development and writing web services, lack of leadership support, etc. Furthermore, since service discovery is by definition an act of finding resources not under the control of the searcher, there is no way (short of re-factoring each discovered service) to force or otherwise cause other service providers to use an ontology (or to adhere to a specific ontology even if they do). The question then becomes: is there some way to realize the benefits of Semantic Web technology for rich discovery of services without having the services based on a formal ontology? Fortunately for a certain class of services there is a way to do this.

We are proposing a different approach for dynamic web service discovery that takes advantage of the structure inherent in web services that are defined by WSDL documents. Since the structure is usually based on XML Schema, there is enough information present in these documents to develop a broadly applicable approach. Furthermore, if a consistent and detailed naming convention of schema artifacts is followed (and recommendations for such are presented in this paper) then discovery can be made more precise. Such an approach can be viewed as a path to richer semantics.

The format of this paper is the following. In section 2, we describe the general approach of our discovery mechanism, and provide a view of the user interface. In section 3, we discuss the application of our approach in the larger picture of an architecture and implementation that combines many service layers and research pieces. In section 4, we

provide our initial evaluation results. Finally, in section 5, we briefly discuss some issues and look toward the next steps.

2 Approach

In previous work, we have demonstrated the effective use of ontologies as a means of performing data integration [10-13] and service chaining [14, 15].

This work extends those previous efforts into the area of web service discovery and assembly. Our approach is as follows:

1. XML Schema definitions of web service input and output items are extracted from a body of WSDL documents.
2. A registry is created that contains the XML Schema definitions, message structure, and the operations of each WSDL file, along with a small amount of metadata.
3. The schema elements are efficiently encoded as a graph set to enable fast lookup as part of the search.

Service discovery is intended to allow subject matter experts to locate information services that provide information needed for some unforeseen task. The assumption is that searchers will be knowledgeable about the domain and somewhat knowledgeable about services, although they may rely on software tools to perform the actual integration (a method for which will be described later in this paper). It is not expected that they will be experts in logic, ontologies, or even search technology.

The mechanics of searching breaks the search terms into three separate parts. The terms are hierarchical. The top level is the overall topic of the search and is usually broad in scope (e.g., sports, medicine, airplanes, etc.). The topic is usually assigned as metadata when the service is registered. Ideally the topic is selected from a pre-set list so that services covering the same subject will use the same topic description, allowing a searcher to use something such as a pull-down list to pick the topic (and thus simplifying the search task). The advantage of this approach is the pre-filtering or ranking of services for locating the type of service being sought (as well as help in disambiguating terms). The disadvantage is the extra bookkeeping that must be done by the registry owner to select the topics and keep them consistent.

The next level is the subject: that is, a term (or terms) for the actual thing being sought. This is a free-text field, and is matched against the text labels of the XML elements of the messages that are in the registry. Various match techniques are used to catch partial or imperfect matches (CamelCase parsing, partial term matching, term expansion using a thesaurus or semantic data model). This approach is sound insofar as the thing being sought is an actual or measurable thing, and not an abstract thing.

The bottom level consists of terms describing the properties or attributes of the subject. These are also matched to the names of elements and attributes of the schemas that make up the messages within the WSDL files.

The three levels of terms can be thought to constitute a very primitive ontology, and the process of searching for matching WSDL messages is similar to a graph-based ontology matching attempt. In fact, the XML structure of the messages in our approach are encoded using a graph-encoding technique similar to the one described in Ait-Kaci [16].

The heart of the search method is in matching the subject search terms to WSDL operations. The other two matching operations (topic and properties) essentially modify the scores of the items returned by the subject search. The assumption is that a searcher will provide a subject, but may not always provide a topic or properties. When these are not provided, the score remains unchanged.

2. 1 Subject Search Method

The following describes our overall subject search method. We index these for readability.

(1) The search software is designed to support the use of multiple multi-word search terms, where each term is separated by a comma (e.g., “satellite status, owner”). The first step is to take the search phrase and turn it into an array of search terms, where multi-word terms are also combined into a single word. So the subject search “satellite status, owner” becomes the string array [“satellite”, “status”, “satellitestatus”, “owner”]. The reason for creating the combined word is that XML schema designers often use “camel case” to name nodes (i.e., elements) in the XML message structure. The idea is that if a combined term happens to be found in a message node name, then that increases the likelihood that it constitutes a good match, and that node’s corresponding operation receives a higher score.

(2) The terms in the string array formed in step (1) are expanded. Each search term (including multi-word terms) are checked against a known library of synonyms, abbreviations, and acronyms. Thus for example “frequency” is expanded to include “freq”, “point of contact” now includes “poc”, and “space object” is expanded to include “satellite”. The expanded array terms are also pluralized. This forms the final array of search terms.

(3) The following scoring loop is then executed. For each search term in the array formed in step (2):

(3.a) The term is looked up in the service registry index, and if found then its corresponding list of operation nodes is returned. The operation nodes are stored as a dot-separated set: WSDLName.OperationName.NodeName

(3.b) For each operation node in (a):

i. The operation node is split into the WSDLName.OperationName (a ‘key’) and the NodeName (the value)

ii. If the key doesn't already appear in the set of answers, then it is added with a beginning score of zero.
iii. The NodeName is split into its component CamelCase parts.
iv. The score is then computed based on the number of terms the NodeName is split into, where the score is 1/number of terms unless the matching term is the last one of the camel case terms; in which case the score is 0.8. For example, the NodeName SatellitePayloadStatus is split into 3 terms, so the score for that operation would be 1/3 if the search term were 'satellite' or 'payload', and .8 if it were 'status'. Note that if a NodeName consists of a single term, then the score is 1 (the highest possible) if the search term matches it. The last word in a camel case term is considered more significant since it tends to be the most significant (i.e., more "noun-like"), whereas preceding words are more adjectival. This score is a called the "occurrence score" – the score for that occurrence of a particular operation. Also note that the same service operation can be encountered multiple times as steps i – iv are repeated.

(3.c) For each operation, the occurrence scores are added up to compute a "term score" for that operation, in the following manner:

$$Term\ Score = \sqrt{\#\ occurences} + Wt \times Avg\ Occurrence\ Score - \log(1 + Occurrence\ Descent\ Level) \quad (1)$$

where:

 # occurrences is the total number of times that operation occurs,

Avg Occurrence Score is the average of all the occurrence scores (i.e., $\frac{\sum occurrence\ scores}{\#\ occurrences}$),

Wt is a weighting factor,

Occurrence Descent Level is the integer count of how far down the highest occurring NodeName is from the root parent node of that operation.

This scoring approach seeks to preserve a balance between how many times a search term appears throughout the named labels of a service operation, how significant that term appears to be, on average, in those labels, and how important the nodes are by computing how far down the message hierarchy they are.

(3.d) The term scores are then added up for each search term. The result is a subject score for each operation.

2.2 Property Score Method

The property scores are computed in a similar manner as the scores for the subject. The main differences are that (1) the property score for each operation is scaled by the number of properties present in that operation (e.g., if a search contained 3 properties, and an

operation contained 2 of them, then the property score for that operation would be multiplied by 2/3), and (2) only operations that are returned as part of the subject search are considered – so operations that match at least one property but not the subject terms are thrown out. For each operation the property score is then added to the subject score.

2.3 Topic Score

The topic of a WSDL is kept as a metadata item in the service registry. The way this item is included in the search is via a pulldown list which contains all of the topics in the registry. This removes the need for the searcher to try and guess what the topic is. Operations matching the topic keep their scores; those with a different topic have their scores reduced, so that they are kept but with a lower ranking.

One of the main reasons for including the topic is to provide some contextual filtering of the other terms. For example, a search for "tank" and "weight" with a topic of "armored vehicles" will score a service operation about Army vehicle features higher than a service about fuel tank capacity under the topic of "Logistics".

2.4 User Interface

Figure 1 depicts our user interface, which is still evolving. It displays four kinds of information: 1) In the upper left of the figure, users can type in *Topic*, *Subject*, *Attributes* to search for and discover Web services of prospective interest to them; in the example, the user has entered "space object" in the *Subject* field. 2) In the upper middle of the figure, the search returns the service operations. We focus on the highlighted "GetSpaceObjectCapabilities".

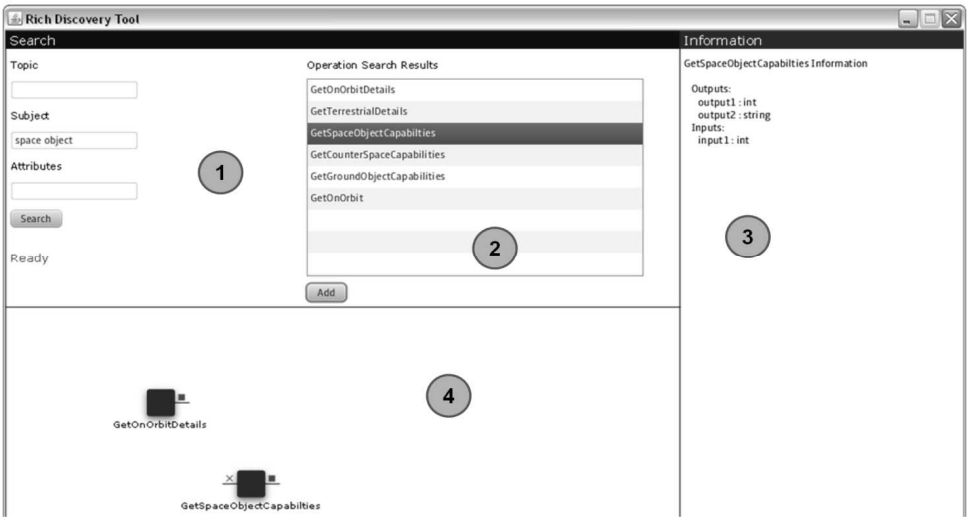


Fig. 1. Service Discovery Graphical User Interface.

3) In the upper right of the figure, details of the service operations are available: in this case, the inputs and outputs of "GetSpaceObjectCapabilities", which includes their data types. 4) Finally, in the lower left of the figure, operations are displayed as boxes that can be connected to form a chain of services via drag-and-drop.

3 Application

We are integrating the service discovery mechanism into a larger effort at MITRE focused on Command and Control (C2), called Composable Capabilities on Demand (CCOD). CCOD intends to provide the capability to rapidly customize virtual systems based on the mission and threat of the day, promoting local innovation by leveraging the use of layered architectures and global integration based on loose couplers at all layers of the architecture. CCOD consists of over 20 individual research projects all providing some capabilities. There are a number of mission scenarios under CCOD. We are involved in a vignette supporting the US Department of Defense Combatant Command (COCOM) information-sharing and interaction to supply assistance to a population experiencing a natural disaster such as the Haiti earthquake of January, 2010. However, for our initial research, because the existing WSDL files that were available were C2-based, we used those, intending to work out a general method for service discovery that could be used for logistical and aid-support operations, when web services for those were more numerous.

4 Evaluation

Currently our evaluation is focused on subjective measures. Typical objective measures, such as recall and precision, slightly modified to incorporate partial or whole aspects of our approach, remain incomplete, requiring a larger population of web services than we have acquired so far.

At a recent CCOD Integration Event in June, 2010, a combined activity for all the research projects involved in the CCOD initiative, we had five people use our tool to discover appropriate services based on several search questions they were given. The questions were to find services that provided:

- The length of a runway
- The callsign and fuel code of a tanker aircraft
- An aircraft's fuel capacity
- The departure base of a tanker aircraft
- An aircraft's tail number and home base
- The payload status and owner of a space object

This gave us 5 people times 6 questions or 30 total test cases. However, some of the search questions turned out to be ambiguous or had multiple correct answers, so we had to throw a few of them out. After this reduction we ended up having a total of 21 person/questions. We compared our approach to that of a generic UDDI type interface, and the results were:

Our approach: 20/21 successful searches
UDDI: 10/21 successful searches

The testers all were very complimentary of our system's search tool and its ability to present results in an understandable manner. A couple, though, expressed some skepticism about web service search and discovery in general.

Concerning our approach in terms of estimating recall, we found this to be really subjective, i.e., deciding which terms to test for. In Table 1 below are displayed the 31 terms which represent the recall terms, and two numbers: how many WSDL files were returned and how many WSDL files should have been returned (as hand determined by us, knowing the domain). The overall recall performance was 162 of 189 returned WSDLs or 85.7%, which is respectable but not great. One issue that caused us problems is the fact that services about aircraft were not returned when the search terms "airplane", "fighter" and "bomber" were used. Although they were clearly implied (and therefore should have been found) in the service, the necessary term matches were not present in the synonym library. Precision measured 162 / 165 or 98%. Of potential significance is our ability to raise recall without reducing precision, which we attribute to using at least weak semantics and the graph structuring.

Table 1. Recall Term Counts (actual vs. expected)

Recall Terms	
aircraft: 12/13 platform: 2/4 satellite: 3/3 mission: 16/16 airplane: 0/2 airframe: 1/4 track: 4/4 runway: 4/4 callsign: 9/10 status: 18/18 tail number: 2/3 payload: 3/3 home base: 1/4 fuel: 8/8 fuel capacity: 3/4 weapon: 7/7	position: 9/10 location: 16/16 route: 1/2 vehicle: 3/3 airspace: 4/6 target: 6/6 tanker: 2/4 fighter: 0/2 bomber: 0/2 airfield: 3/4 plan: 5/6 ato: 7/7 equipment: 7/7 jammer: 3/3 emitter: 1/4
Recall: 162 / 189 (85.7%) Precision: 162/165 (98%)	

5 Discussion and Future Research

Although our approach described in this paper uses only weak term semantics projected from the actual WSDL structures, when combined with heuristics used by the algorithm (different weights for topic, subject, property, and use of the hierarchic structure of the WSDL), and a simple lexicon of synonyms and acronyms to assist in query expansion, it promises to assist technically unsophisticated users at discovering relevant Web services, and then enabling their drag-and-drop composition – all without the use of a sophisticated domain ontology. We do believe that more sophisticated terminologies and domain-specific ontologies will enable more accurate search and discovery methods, and will draw upon these as they become available. For example, some early Community of Interest (COI) vocabularies are emerging within the DoD for command and control, logistics, etc., and some with associated ontologies to represent the term meanings. But in general, these are not yet emerged. Also, some external lexical-conceptual resources such as WordNet can be employed and integrated into our approach, but in general, we have found these to not have the term specificity and complexity of domain knowledge that we need. Hence, we view our technical contribution as enabling a transition strategy between

purely syntactic (keyword-based) methods and much richer semantic methods using ontologies, to assist users and developers over the near and mid-term. In addition, we believe our method advances the state of the art and can be expanded on when richer resources become available.

Concerning structural issues, WSDL and XML files (and WSDL's syntax is XML-based) are tree-based, since XML's underlying data model is tree-based. So in our design and implementation of an efficient representation for the structure of WSDL services, we focused on a tree-based representation, though we did plan for a graph-based generalization, and our encoding scheme reflects that. We considered various graph-based representations and efficient encodings of subsumption reasoning based on those representations. These include considerations from early bit encodings such as [16-21] to more recent work that includes prime number encodings, such as [22-23].

We are also very interested in addressing more objective evaluative measures such as typically captured by recall and precision metrics (sufficiently adjusted to fit the circumstances). However, our total set of WSDL services is really not large enough yet to apply these measures meaningfully. The set of structure-based WSDL-defined Web services will undoubtedly grow in the near term, given that ontologies that define more precise semantics for domains of those services unfortunately will continue to be lacking. So we think therefore that we will have more service definitions to work with in the future, and hence be able to evaluate our discovery methods more precisely against that larger set.

Finally, we have identified several recommendations and conventions for service schema developers to adopt, to increase the value of the syntactic and structural services they define when using a light semantic approach for discovery such as ours, but which will also benefit approaches with richer semantics: 1) use descriptive names for elements and attributes (FuelStationLocation) and avoid general terms ("items", "response"); use whole words and avoid contractions ("track" not "trk"); follow a consistent style for CamelCase; and design schemas with as much specificity as possible (e.g., avoid xs:any, xs:all). These practices, when combined with a service-discovery approach such as ours, can help find and reuse web services – until richer ontologies and lexical resources are available, and until developers become more sophisticated with semantic technologies.

Acknowledgments. The views expressed in this paper are those of the authors alone and do not reflect the official policy or position of The MITRE Corporation or any other company or individual.

References

1. Baumgartner, R., Flesca, S., Gottlob, G.: Visual Web Information Extraction. VLDB Conference, 2001

2. Kourtesis D., Paraskakis I.: Combining SAWSDL, OWL-DL and UDDI for Semantically Enhanced Web Service Discovery. In Bechhofer S. et al.(Eds.): ESWC 2008, Lecture Notes in Computer Science 5021, Springer-Verlag Berlin Heidelberg, pp. 614-628 (2008)
3. Kourtesis, D., Paraskakis, I., Friesen, A., Gouvas, P., Bouras, A.: Web Service Discovery in a Semantically Extended UDDI Registry: the Case of FUSION. In: Camarinha-Matos, L., Afsarmanesh, H., Novais, P., Analide, C. (Eds.) IFIP International Federation for Information Processing, Establishing the Foundation of Collaborative Networks, vol. 243, Springer, Boston, pp. 547-554 (2007)
4. Web Service Execution Environment, http://www.wsmx.org/
5. Service-Finder, http://www.service-finder.eu/
6. Mokhtar, S. B., Kaul, A., Georgantas, N., Issarny, V.: Efficient Semantic Service Discovery in Pervasive Computing Environments. Proceedings of the ACM/IFIP/USENIX 2006 International Conference on Middleware, IFIP International Federation for Information Processing, Melbourne, Australia. M. van Steen and M. Henning (Eds.): Middleware 2006, Lecture Notes in Computer Science 4290, pp. 240–259 (2006)
7. Mokhtar, S. B., Preuveneers, D., Georgantas, N., Issarny, V., Berbers, Y.: EASY: Efficient semAntic Service discoverY in pervasive computing environments with QoS and context support. The Journal of Systems and Software 81, pp. 785–808 (2008)
8. Web Services Description Language (WSDL), http://www.w3.org/TR/wsdl
9. Business Process Execution Language (BPEL), http://en.wikipedia.org/wiki/Business_Process_Execution_Language
10. Obrst, L., Stoutenburg, S., McCandless, D., Nichols, D., Franklin, P., Prausa, M., Sward, R.: Ontologies for Rapid Integration of Heterogeneous Data for Command, Control, & Intelligence. Chapter in: Obrst, L., Janssen, T., Ceusters, W. (eds.) Ontologies and Semantic Technologies for the Intelligence Community, pp. 71-89. IOS Press, Amsterdam, The Netherlands. IOS Press book series: Volume 213 Frontiers in Artificial Intelligence and Applications (2010)
11. Samuel, K., Obrst, L., Stoutenberg, S., Fox, K., Franklin, P., Johnson, A., Laskey, K., Nichols, D., Lopez, S., Peterson, J.: Applying Prolog to Semantic Web Ontologies & Rules: Moving Toward Description Logic Programs. The Journal of the Theory and Practice of Logic Programming (TPLP), Massimo Marchiori, ed., Cambridge University Press, Volume 8, Issue 03, pp 301-322 (2008)
12. Obrst, L., McCandless, D., Stoutenberg, S., Fox, K., Nichols, D., Prausa, M., Sward, R.:. Evolving Use of Distributed Semantics to Achieve Net-centricity. Regarding the “Intelligence” in Distributed Intelligent Systems, AAAI Fall Symposium, Arlington VA, Nov. 8-11 (2007)
13. Stoutenberg, S., Obrst, L., Nichols, D., Franklin, P., Samuel, K., Prausa, M.: Ontologies and Rules for Rapid Enterprise Integration and Event Aggregation. Vocabularies, Ontologies and Rules for the Enterprise (VORTE 07), EDOC 2007, Annapolis, MD, Oct. 15-19 (2007)
14. McCandless, D., Obrst, L.: Using Ontology Alignment to Dynamically Chain Web Services. Ontology Matching Workshop, poster, International Semantic Web Conference (ISWC) 2009, Oct. 25-29, Chantilly, VA (2009)
15. McCandless, D., Obrst, L.: Dynamic Web Service Chaining using OWL and a Theorem Prover. Third IEEE International Conference on Semantic Computing, Berkeley, CA, USA - September 14-16 (2009)
16. Ait-Kaci, H., Boyer, R., Lincoln, P., Nasr, R.: Efficient Implementation of Lattice Operations. TOPLAS 11-1 (1989)

17. Ait-Kaci, H.: A Lattice-Theoretic Approach to Computation Based on a Calculus of Partially-Ordered Type Structures. Ph.D thesis, Computer and Information Science Dept., Univ. of Pennsylvania, Philadelphia, PA (1984)
18. Caseau, Y., Habib, M., Nourine, L., Raynaud, O.: Encoding of Multiple Inheritance Hierarchies and Partial Orders. Computational Intelligence 15 (1), pp. 50-62 (1999)
19. Fall, A.: Heterogeneous Encoding. In Proceedings of International KRUSE Symposium: Knowledge Retrieval, Use, and Storage for Efficiency, Gerard Ellis, Robert Levinson, Andrew Fall, Veronica Dahl, eds., Santa Cruz, CA, Aug. 11-13, pp. 134-146 (1995)
20. Hellerstein, J.M., Naughton, J.F., Pfeffer, A.: Generalized search trees for database systems. In: Proceedings of the 21st International Conference of Very Large Data Bases, VLDB'95 (1995)
21. Krall, A., Vitek, J., Horspool, N.: Near optimal hierarchical encoding of types. 11th European Conference on Object Oriented Programming (ECOOP'97). Springer (1997)
22. Preuveneers, D., Berbers, Y.: Prime numbers considered useful: Ontology encoding for efficient subsumption testing, Tech. Rep. CW464. http://www.cs.kuleuven.be/publicaties/rapporten/cw/CW464.abs.html. Department of Computer Science, Katholieke Universiteit Leuven, Belgium (October 2006).
23. van Bommel, M. F., Wang, P.: Encoding multiple inheritance hierarchies for lattice operations. Data & Knowledge Engineering, Volume 50, Issue 2, August, pp. 175-194 (2004)

Using Ontological Information to Enhance Responder Availability in Emergency Response

Paul Ngo[1] and Duminda Wijesekera[1],

[1] Department of Computer Science, George Mason University,
Fairfax, VA 22030
{pngo1, dwijesek}@gmu.edu

Abstract. Ensuring effective communications during emergencies is an important issue for any functional government. One way to address this issue is to ensure the availability of the key personnel capable of making the appropriate decisions and taking timely actions with sufficient resources. Many XML-based languages such as the Emergency Data Exchange Language (EDXL) and associated Common Alert Protocol (CAP) have been designed to provide a basis for such communications. To ensure that messages are delivered in a timely manner, we propose some role and task based ontological enhancements for these languages. We show by example how the ontological enhancements can be used to enhance availability of emergency personnel in case of a need.

Keywords: Emergency Availability, Emergency Ontology, Emergency Response.

1 Introduction

Multiple mega-scale emergencies highlight the need for better global emergency response. The September 11th 2001 terrorist attacks in New York, Indonesian Tsunami in 2004, Hurricane Katrina in 2005, Sichaun earthquake in 2008, and the Haiti earthquake and Pakistani floods in 2010 are examples of a few. During these emergencies, urgent task-related communications must reach key officials in a timely manner. Emergency responders must know how to contact the person in charge of a specific task, which is sometimes difficult due to not being able to locate a telephone number, or when reached using directory information, the person may not be available or may have been reassigned to a different job/task. There is no automated method of redirecting the call to the current person who should be attending to that task and is on-duty at the time of the call. It's preferable to have a subject, task specific, 911-like calling number for each task, time and locality. The objective of this research is to reach such a capability for the real-time needs of emergency responders.

The basic 911 services provided in the USA serve as a pseudo name that is available to the general public at every time and every location, but is mapped to a collection of numbers belonging to an emergency call center based on the call originator's location. Although we take it for granted, the public switched telephone

network (PSTN) has been designed to translate the pseudo name 911 to a location specific telephone number. Thus this address translation depends only on a single parameter, *the caller's location*. Our objective is to extend this capability in order to facilitate the communication beyond the first call from the public. The issue of extending this paradigm for emergency responders to contact each other depends on a plethora of parameters, nature of the emergency, priority of immediate needs and resources to fulfill them. We agree that if a person is not available to receive the request, the communication breaks down. But often, locating this person takes multiple calls/SMS and email messages before the correct person can be reached. It is this gap that we propose to fill by developing an ontology (hence the lexicons) as the need to parameterize the basic 911 service.

With support from the Department of Homeland Security Disaster Management eGov Initiative, the Organization for the Advancement of Structured Information Standards (OASIS) technical committee on emergency management developed a set of standards for the interagency exchange of emergency management data and messaging [1,2,3]. Standards [1] and [2] developed the Emergency Data Exchange Language (EDXL) that provides a set of XML based tags to exchange the information needed to handle an emergency. To route, receive and respond to these messages, the responder anticipating an emergency duty related request must be identifiable by other collaborators that will need his services. Consequently, our development enhances the EDXL entities to ensure that the calling party is able to reach the *best called party* based on the *latter's availability*. To do so, we propose that all potential responders expose their *capabilities* and fallback options in case they cannot be reached during emergencies. These capabilities of responders include the role played in an organization, the tasks the actor can execute, estimated time to respond to a request (perhaps due to many emergency calls) or execute these tasks, available resources, direct contact information, and an alternative contact chain in case of unavailability of the best contact and the sensitivity of the information authorized to receive, and a contact to report complains about the quality of service, including contacting difficulties.

The rest of the paper is written as follows. Section 2 describes related work. Section 3 describes the linguistic abstractions proposed in EDXL and its messaging language. Section 4 proposes our enhancements to ensure the availability aspect of the actors. Finally, section 5 describes our concluding comments. Further details are available in a Technical Report at [11].

2 Related Works

In recent years, there have been a number of publications on building ontologies to solve different aspects of emergency handling. We discuss a few that are considered to be relevant to our work. Li et al. [5] proposes an ontology for crisis management. Although they defines a common set of vocabularies that can be used to facilitate an effective communication, they do not address failure scenarios in reaching key responders in a time of crisis.

Yu et al. [8] illustrates a good use of Activity-First Method (AFM) proposed by Mizoguchi [10] to construct an emergency ontology for creating a decision support system from existing emergency documents and use cases. This methodology is aimed at decomposing the emergency documents into data components for further integration based on emergent incidents. Although this emergency ontology helps decision makers sort out existing knowledge and reach critical decisions faster and more efficiently, it does not address how to ensure the availability of decision makers during an emergency.

Malizia et al. [6] constructs an emergency ontology for event notification and system accessibility. Using the knowledge that reflects users' needs, ways to present their needs, the nature of the emergency and available technologies makes it possible to reach more people. To build such a complex ontology, the authors use three domain concepts: accessibility, user profiles and devices and verification of the validity and integrity of knowledge by using first order logic. Although the proposed ontology may address the information needs for sharing and integrating emergency notification messages and provide the accessibility for different kinds of users under different conditions, it does not address the information needs for ensuring the responder's availability at the time of the need.

The open ontology approach [9] provides great flexibility to extend into a mission-oriented ontology. In order to do so, an open ontology provides multiple spaces and views that must be taken into account during the design phase. It also provides a theoretical approach to build such an ontology rather than providing a practical open ontology for emergency response. To the best of our knowledge, no one has extended this concept and developed it into a practical open ontology yet.

To facilitate sharing of information across all levels of government, the Federal Government has initiated the Universal Lexical Exchange (ULEX), which helps define the top sharable objects that can be formed into a coherent message that can be validated via the XML schema [17]. Although ULEX defines sharable contact information, the objective is to provide the contact information for deployable systems and services, and not the availability of the contact person during an emergency based on the person's job description. Universal Core (UCore) is another Federal information sharing initiative that supports the national information sharing strategy among all federal departments and agencies. UCore defines an implementable specification in XML schema that enables the information sharing of well-known and comprehensible concepts of *who, what, when and where* [18]. Although these concepts can address some aspects of information sharing for emergencies, they do not addressing how the contact would be used to locate the person during an emergency.

The US Federal Government has established a Government Emergency Telecommunications Service (GETS) program [14], which ensures a high probability of call establishment during a crisis when the PSTN is congested. This program provides a specific and recognizable phone number to obtain a higher priority for establishing a call. In recent years, with the increased prevalence of wireless phones, the Federal Government established a Wireless Priority Service (WPS) [15] program, where subscription information is used to identify high priority callers. However, both GETS and WPS services do not guarantee call establishment but rather provide best effort due to the network bandwidth availability. These services are considered

complementary to our work on ensuring updated status is maintained regarding the availability of the responder or his alternate.

Many standards have been developed by OASIS that have been widely adapted in data communication for emergency handling. One of the recent standard releases is the Common Alerting Protocol (CAP) [12, 13], which is the primary communications protocol for exchanging emergency alert messages between different parties. CAP has been used, implemented and deployed by a number of agencies and firms [16]. In this paper, we enhance CAP by adding necessary elements into the CAP schema to enhance reaching the responders in an emergency. We also illustrate the use of these elements in a real life emergency scenario.

Last but not least is the EDXL language, which was developed by OASIS and became a standard in 2006 [1]. We strengthened the EDXL language by adding syntax that can be used to attempt to deliver messages to emergency personnel when the existing mechanisms fail.

3 EDXL

EDXL is a language designed for sharing information and exchanging data among local, state, tribal, national and non-governmental organizations to facilitate emergency response [1]. Figure 1, taken from Page 10 of [1], shows the entities used in creating the EDXL syntax in the form of an Entity Relationship (ER) model, where the entity in red is our enhancement that will be described in Section 4.

As Figure 1 shows, at the highest level, each EDXL distribution element (i.e. message) has six required attributes and six optional attributes. In addition, every message has a target area identifying a geographical region and a content object describing the incident, confidentiality levels and roles for the originator and consumer of the message.

Required attributes of the distribution element consist of a distribution ID, sender ID, date and time the message was sent, distribution status (consists of one of the four values: Actual, Exercise, System and Test), a distribution type consisting of value such as Report, Update, Request, Sensor Status, etc., and Combined Confidentiality having the most restrictive level of confidentiality sought for the combined payload.

The optional attributes consist of the language used in the message and (possibly multiple instances of) the sender's role, recipient role, keywords, distribution references (indicating distribution constraints) and possibly an explicit address for delivery. The explicit address is an XML schema.

EDXL messages can have four kinds of *optional* roles. They are sender's role, recipient's role, originators' role and consumers' role. These roles are supposed to be used for two purposes: (1) identifying potential recipients and (2) message distribution. In addition, explicit addresses can also be used for the latter task. The recommended usage syntax for the sender ID is *actor@domain-name* (such as dispatcher@example.gov) where the domain-name is guaranteed using the Internet Domain Name System.

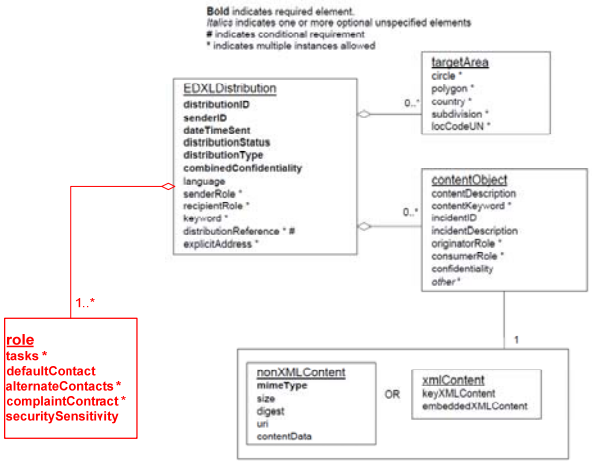


Figure 1: EDXL-DE Entity Relationship Diagram

4 Enhancing EDXL for Responder Availability

Before we explain our enhancements, several comments are worth mentioning. First, EDXL and CAP messages were designed for multiple purposes such as human-to-human, machine-to-human and machine-to-machine communications, etc., as shown by the fact that distribution element consists of optional fields such as *Sensor Status*, etc. For sensors, attributes like *roles* do not apply, but they do for human responders to emergencies. For example, we want to identify the Paramedic in an emergency response team (the role, but not the person) and his capabilities (such as is he authorized or trained to execute a certain type of medical routine like cardiac resuscitation, etc.?). Thus, for human responders, the role is more central than the recipient address, and the tasks that he is able to execute in that role. Therefore, in our enhancement the role is a mandatory attribute (marked by 1-* in Figure 1).

Because our objective is to enhance reaching the human responders with most suitable capabilities, we need to consider failure modes. One of the most important issues of recipient-address based emergency messages is that if that recipient is unreachable then it becomes the sender's responsibility to find the next available responder. Also any delivery system, such as an automated phone dispatcher, pager, SMS or email system should have an inbuilt mechanism to redirect the message automatically to the next appropriate responder. In order to facilitate this capability, either using an automated redirecting algorithm or in a sender initiated system, we propose creating a lexicon/ontology that has a list of alternative roles (where the role to person/phone number/IP address will be automated). In order to address the failure of these alternatives, we specify a *complain* role that should deliver the message to the higher authoritative personnel.

The redirecting algorithm can be easily implemented in the Private Branch Exchange (PBX) of the caller. [19] describes three common failure scenarios, Callee

Busy, Callee Unanswered or Global Errors. In all cases, when the call cannot be connected as dialed, the caller Sessions Initiation Protocol (SIP) gateway sends a *disconnect* message with the appropriate error code to the caller's PBX. Before this message is sent to the caller, we can inject the redirection mechanism by providing the PBX with a list of the default, the alternative and the complaint numbers, as will be shown in the algorithm depicted in Figure 2. For this to work properly, we made two assumptions. First, we assume that the local PBX has an Emergency Address book that is capable of translating the list of tasks to the local numbers based on their relevancy. Figure 7 illustrates an example with the <role> and <tasks> tags. Second, we assume that the order of relevancy can be selected by the local PBX. For example, in Louisiana, floods have more priority than earthquake. However, in California, the order must be reversed. This way, the selection algorithm can be regionalized, For now, we assume that our sorting algorithms addresses this based on its locality although we are working on separating these concerns. The PBX first makes a call to the *defaultContact*. If the PBX receives the Disconnect message from the local SIP gateway, the PBX will redirect the call to numbers on the alternative list. If there are no more alternatives, the PBX will redirect the call to the complaint number. Figure 2 depicts the pseudo-code for the algorithm that can run as an application at the PBX and make repeated attempts to facilitate availability of responders.

Roles and Tasks	Other Contacts	Contacts
Role: Emergency Gas technician **Tasks:** (1) Licensed to shut down main valves, (2) (dis)connect household lines, (3) Repair valves **zip codes** 22222, 22221	Email: emergency@gasexpert.com SMS: 7031111111 Response Window: 24 hrs/day Estimated Response Delay: 20 seconds	Default: 7031111111 Alternatives: 7032222222 7033333333 Complaint: 7039999999
Role: Emergency Gas technician **Tasks:** (1) Licensed to shut down main valves, (2) (dis)connect household lines, (3) Repair valves **zip codes** 22222, 22221, 22204, 22223	Email: emergency@gassol.com SMS: 7031110001 Response Window: 7AM to 10PM EDT, weekdays 9AM – 6PM EDT, weekends Estimated response Delay: 15 minutes	Default: 7031110001 Alternatives: 7031110002 7031110003 Complaint: 7031110005
Role: Emergency Gas technician **Tasks:** (1) Licensed to shut down main valves, (2) (dis)connect household lines, (3) Repair valves **zip codes** 22222, 22201, 22204, 222205	Email: emergency@gaspro.com SMS: 7032220001 Response window: 6AM – 11PM EDT, weekdays 8AM – 10PM EDT, weekends Estimated Response Delay 10 minutes	Default: 7032220001 Alternatives: 7032220002 7032220003 Complaint: 7032220009

Table 1: Key Words Translation

Figure 2 illustrates a pseudo code redirection algorithm at the local PBX. The *makeEmergencyCall* method accepts one parameter of the role node, which has been populated with the tasks that are relevant to the emergency. The *getTableFromRoleAttrs* method is then called to retrieve the table of contacts by searching the Emergency Address book for the contacts that are associated with the tasks. The table is then sorted based on time and the relevancy. The best matched entry in the table is then added to the role in three separate tags: defaultContact, alternateContact and complaintContact. The defaultContact is then called. If the disconnect is received from the local SIP gateway, each of alternateContacts is then called. If every call to the alternateContacts is failed, the complaintContact is called.

```
public int makeEmergencyCall (Node role)
begin
  table = getTableFromRoleAttrs(role.getTasks())
  role = sort(table, getCurrentTime(), role)
  defaultContact = getDefaultContact (role)
  returnCode = dial(defaultContact)
  if (returnCode == Disconnect)
  begin
    listAlts = getAltContact(role)
    while (listAlts is not empty)
    begin
      altContact= getNextAlt(listAlts)
      returnCode = dial(altContact)
      if (returnCode == OK)
        break
    end
    if (returnCode == Disconnect)
    begin
      compContact = getComplaintContact(role)
      returnCode = dial(compContact)
    end
  end
   return returnCode
end
```

Figure 2: Local PBX Redirection Algorithm

```
<role>
<tasks>
<valueListUrn>valueListURN</valueListUrn>
<task>value<task>
<estimatedTimeToFinish>time</estimatedTimeToFinish>
<currentResponseDelay>time</currentresponseDelay>
</tasks>
<defaultContact>contactURN</defaultContact>
<alternateContacts>
<valueListUrn>valueListURN</valueListUrn>
<contact>value</contact>
</alternateContacts>
<complaintContact>valueListURN</complaintURN>
<securitySensitivity>Security classification level
</securitySensitivity>
</role>
```

Figure 3: EDXL Enhancement

4.1 Ontological Enhancements or Roles

In the current EXDL-DE specification, a *mandatory* recipient role is given as a list of structures where each element is a potential recipient.

```
<recipientRole>
    <valueListUrn>valueListUrn</valueListUrn>
    <value>value</value>
<recipientRole>
```

Here the content of <valueListUrn> is the Uniform Resource Name of a published list of values and definitions, and the content of <value> is a string (which may represent a number) denoting the value itself. Multiple instances of the <value> may occur with a single <valueListUrn> within the <recipientRole> container. In addition, the <recipientRole> is *not* a required element. Our enhancements propose the following additions to a role as depicted in Figure 3.

5 Conclusion

We have taken a collection of standards for emergency management messages and proposed enhancements that would ensure that the messages are delivered to a set of recipients that are capable of responding to the needs at hand. Our proposal is based

on a set of attributes that characterize the tasks that are needed of an external emergency handling entity. We have expressed these attributes by extending the proposed EDXL language. Our objective in doing so was to provide a 911 like pseudo name that is parameterized based on the organization, required responder's role and tasks he is expected to perform in order to satisfy the needs of the call. Our ongoing work addresses translating these pseudo names to addresses available on the telephone, email and pager services so that they can take advantage of PSTN based and wireless based priority calling services provided for specified actors of federal, state, local and tribal agencies.

Reference

1. Emergency Data Exchange Language (EDXL) Distribution Element, v. 1.0 OASIS Standard EDXL-DE v1.0, 1 May 2006.
2. Emergency Data Exchange Laguage Resource Messaging (EDXL-RM) 1.0 OASIS Standard incorporating Approved Errata 22 December 2009.
3. OASIS. www.oasis-open.org/committees/emergency.
4. Jacqueline Yang, Duminda Wijesekera and Sushil Jajodia, *Subject Switching Algorithms for Access Control in Federated Databases*, in the proceedings of the 15th Annual IFIP Conference on Database Security, 2002. Pages 61-74.
5. Xiang Li, Gang Liu, Anhong Ling, Jian Zhan, Ning An, Lian Li, and Yongzhong Sha, "Building a Practical Ontology for Emergency Response Systems," *Computer Science and Software Engineering, 2008 International Conference on*, 2008, pp. 222-225.
6. A. Malizia, T. Onorati, P. Diaz, I. Aedo, and F. Astorga-Paliza, "SEMA4A: An ontology for emergency notification systems accessibility," *Expert Systems with Applications*, vol. 37, Apr. 2010, pp. 3380-3391.
7. W. Xu and S. Zlatanova, "Ontologies for Disaster Management Response," *Geomatics Solutions for Disaster Management*, 2007.
8. Kai Yu, Qingquan Wang and Lili Rong; , "Emergency Ontology construction in emergency decision support system," *Service Operations and Logistics, and Informatics, 2008. IEEE/SOLI 2008. IEEE International Conference on* , vol.1, no., pp.801-805, 12-15 Oct. 2008.
9. P. Di Maio, "An Open Ontology for Open Source Emergency Response System", http://opensource.mit.edu/papers/TOWARDS_AN_OPEN_ONTOLOGY_FOR_ER.pdf
10. R.Mizoguchi, M.Ikeda, K. Seta, J.Vanwelkenhuysen, "Ontology for Modeling the World from Problem Solving Perspectives," Proc. of IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing,1995, pp.1-12
11. Technical Reports. http://cs.gmu.edu/~tr-admin/.
12. Common Alerting Protocol, v.1.1, OASIS Standard CAP-V1.1, October 2005.
13. Common Alerting Protocol Version 1.2, OASIS Standard, 01 July 2010.
14. Government Emergency Telecommunications Service. http://gets.ncs.gov/.
15. Wireless Priority Service. http://wps.ncs.gov/.
16. Common Alert Protocol (CAP). http://www.oasis-emergency.org/cap.
17. ULEX. http://www.lexs.gov/content/ulex.
18. Universal Core (UCore). https://ucore.gov/ucore/.
19. Call Flow Scenarios for Calls Failed. http://www.cisco.com/en/US/products/sw/iosswrel/ps1831/products_programming_reference_guide_chapter09186a0080087348.html.

Patient-Centric Secure-and-Privacy-Preserving Service-Oriented Architecture for Health Information Integration and Exchange

Mahmoud Awad and Larry Kerschberg,

Center for Health Information Technology, George Mason University, http://hit.gmu.edu

Abstract. In this paper, we propose a secure and privacy-preserving Service Oriented Architecture (SOA) for health information integration and exchange in which patients are "part owners" of their medical records, have complete ownership of their integrated health information and decide when and how data is modified or exchanged between healthcare providers or insurance companies. This architecture is different from integrated Personal Health Record (PHR) such as Google Health and Microsoft HealthVault in that electronic health records are not stored in online databases but instead are aggregated on-demand using web service requests. Web service providers working on behalf of the patients do not keep copies of the complete EHR but instead provide a pass-through service, and would require PKI-based security certificates to initiate health information exchange.

Keywords: Privacy Ontology, Electronic Health Record, Service Oriented Architecture, Health Information Exchange, HER, SOA, HIE.

1 Introduction

Patient health records (in electronic or paper form) such as medications, lab results and family history are owned by the healthcare establishment that requested or created such records. Even though patients can request copies of their medical records, the process of getting such records is neither streamlined nor convenient. Photocopies of large medical files are costly and in most cases unreadable, and, in the case of electronic systems, these records are usually in proprietary format that are hard to integrate with each other. As more healthcare providers switch to Electronic Health Records (EHR), most of these issues will be overcome but the security, privacy and ownership of these medical records remain hard-pressed issues.

The Health Insurance Portability and Accountability Act (HIPAA), which was enacted in 1996, includes provisions that govern certain privacy aspects related to patients health records. These provisions apply to healthcare providers such as hospitals, physicians and laboratories, but do not apply to companies that aggregate these health records in electronic format such as Google Health, Microsoft HealthVault and Indivo. Most people consider the state of their health to be very

confidential and, therefore, security and privacy concerns may drive people away from such integrated systems in spite of all the strict online privacy policies established by Google and Microsoft. People would rather deal with a healthcare entity that is covered under an enforceable federal law than deal with unenforceable privacy policies established by corporations that have objectives that overshadow and eclipse the confidentiality of an individual's lab results or family medical history.

In this paper, we propose a secure and privacy-preserving Service Oriented Architecture (SOA) for health information integration and exchange. The proposed architecture is different from integrated EHR systems such as Google Health and Microsoft HealtVault in that electronic health records are not stored in online databases but instead are aggregated on demand using web service requests. All health information exchanges have to be approved by the patient and would require one-time use secure tokens for authentication, privacy policies to control data elements exchanged and fine-grained security policies to control data element values exchanged. As a proof of concept, we developed a prototype showing how privacy and security policies are created and how they are applied as part of an EHR exchange.

2 Proposed Architecture

In our proposed architecture, shown in Figure 1, the patient is represented by an application server that communicates with healthcare providers using a set of web services. This application server contains a set of privacy policies and security policies that govern all data exchange requests, and does not have the capability to store the patient's complete health record.

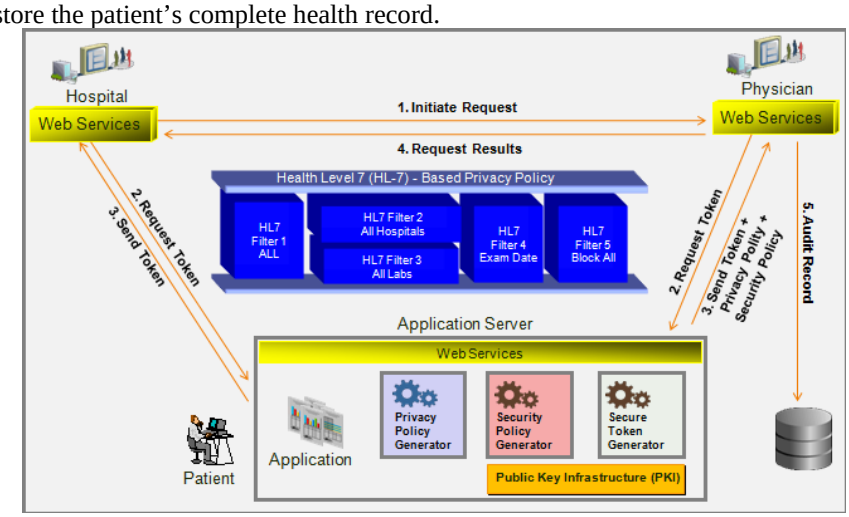


Fig. 1. Patient-Centric Secure System Architecture.

The server representing the patient consists of the following components:

1. Database contains fine-grained historical audit trail of all data exchange requests among healthcare providers, which includes additions, modifications and deletions of health record structure or data. The patient's medical history can be reconstructed using this audit trail but only the patient has privileges to initiate such request.

2. Privacy Policy Generator (PPG) generates privacy policies by defining which data structure elements are allowed to be exchanged between healthcare entities. The policy itself is represented using HL7 CDA syntax and acts as a filter between a web service and its data store. Privacy policies can be generated manually or via templates such as Continuity of Care Record (CCR) which is an HL7 constraint.

3. Security Policy Generator (SPG) generates security policies that restrict records retrieved by a database in response to an EHR query. These security policies enforce fine-grained access and are modeled similar to relational database fine-grained security access control. In order to generate new security policies or modify existing policies, the SPG receives a request from the PPG with a privacy policy identifier, a healthcare provider identifier and the data elements that need to be secured by the new security policy.

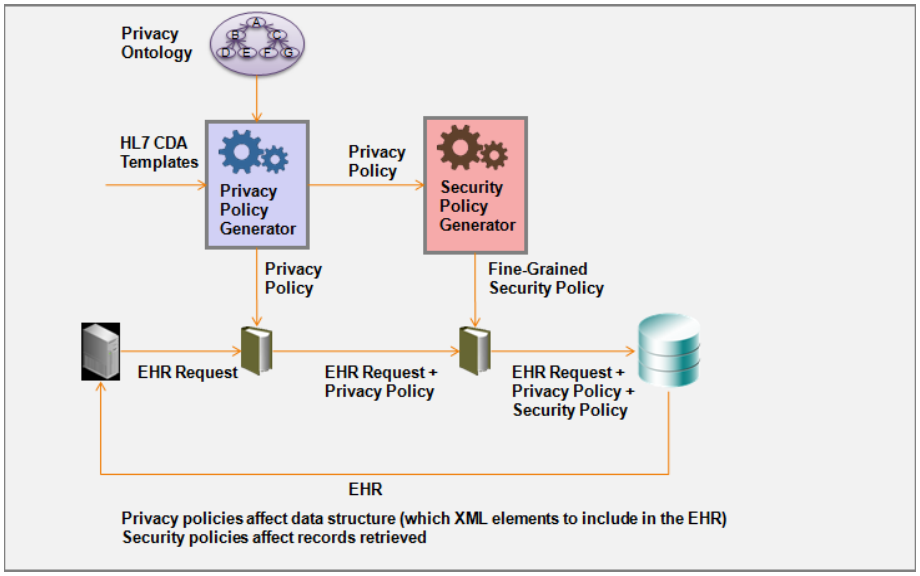


Fig. 2. Privacy and Security Policy Generators.

The architecture offers a clear separation between privacy policies and security policies in order to provide better flexibility in producing and applying the filters and

predicates produced by the PPG and SPG respectively. Privacy filters are applied first to restrict data elements in an XML response (or columns in case of relational tables), then security policies are applied to limit the data element values. Implementation details depend on the architecture of the medical record system implemented internally at the healthcare providers or health insurance companies. Systems that use relational database can use fine-grained access control to implement security policies and systems that use XML databases can use XML schemas to validate the XML document produced.

1. Secure Token Generator (STG), Requests for EHR exchange are initiated but not executed until secure tokens are generated by the STG. The tokens are generated using PKI and use a random number to ensure they are used only once.

2. Privacy Ontology; helps the PPG determine relationships among healthcare providers and between EHR data elements and provides a mapping between the healthcare providers and EHR data elements. Default privacy policy templates are generated using this privacy ontology. An example of relationships between healthcare providers is all the hospitals and medical practices that use Quest Diagnostics as their diagnostic laboratory testing facility. This knowledge simplifies the process of generating security policies that would allow lab results to be exchanged between these medical facilities and Quest Diagnostics. Also, knowing that the patient's primary family physician is a registered practitioner at particular hospital helps establish the level of trust in data exchanges between the physician and various offices within the hospital.

3. Applications are used to: a) Monitor data exchange requests and help the users decide whether to approve or reject a request; b) Produce privacy policies and security policies; c) Query an individual component of the EHR or produce a complete EHR by issuing EHR integration web service requests to all the registered healthcare providers; and d) Review and correct individual components of the EHR by issuing correction requests to the system holding the affected record.

ThePrivacy Ontology is an important component of our architecture and a subject of active research. We are motivated by the HL7 Security and Privacy Ontology (See: http://wiki.hl7.org/index.php?title=Security_and_Privacy_Ontology). The ontology was developed using their methology and use cases dealing with access control based on category of action, of object, of structural role, of functional role, and on multiple role values. Additional use cases deal with facilitating an automatic decision function and the design of an access control system.

The HL7 Security and Privacy Ontology is specified in the Web Ontology Language (OWL) and is implemented in the Protege Ontology Editor from Stanford University. We are presently investigating how to incorporate patient-centric privacy and security authorization constructs into the Privacy Ontology so as to strike a balance between patient privacy, the secure exchange of health information, and mechanisms to ensure the chain-of-custody of electronic health records.

3 Application Prototype

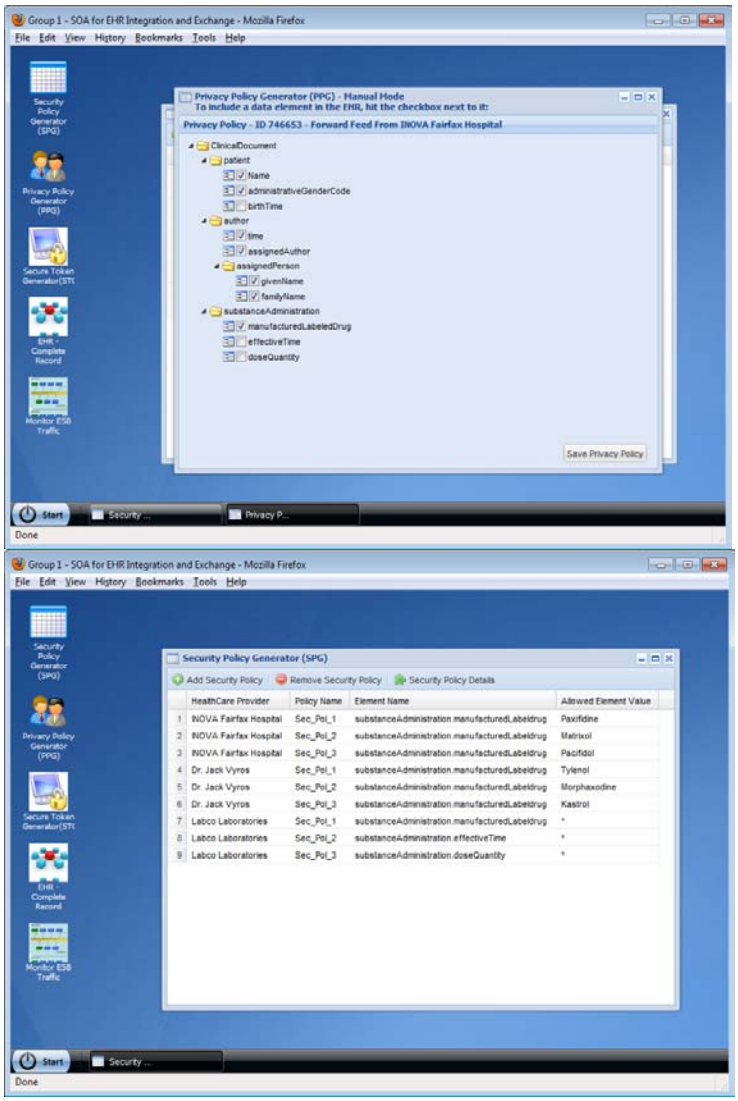


Fig. 3. Application Prototype.

4 Discussion

Any comprehensive solution for EHR integration and exchange has to be technologically feasible but also politically acceptable. Healthcare providers will

always claim ownership of all medical records in their possession, and as long they are HIPAA-compliant, we have to assume that they have developed adequate internal security and privacy policies to protect these medical records. Our proposed solution only requires a web services layer around existing systems while giving patients an active role in the EHR exchange instead of the current practice of providing their healthcare providers with a blank authorization to exchange their EHR with anybody. Also, fully centralized EHR integration solutions are prone to privacy and security lapses and disruptive hacker attacks such as Denial Of Service (DOS). Fully distributed solutions, on the other hand, are prone to data loss if they do not offer proper data redundancy and backup strategies. Our proposed solution maintains the existing distributed network of systems represented by the healthcare providers but offers a secure method for data integration on demand.

5 Conclusion

In this paper, we propose a secure and privacy-preserving SOA for health information integration and exchange in which patients are "part owners" of their medical records, have complete ownership of their integrated health information and decide when and how data is modified or exchanged between healthcare providers or insurance companies. This architecture is different from integrated Electronic Health Record (EHR) such as Google Health and Microsoft HealtVault in that electronic health records are not stored in online databases but instead are aggregated on demand using web service requests. Web service providers working on behalf of the patients do not keep copies of the complete EHR but instead provide a pass-through service, and would require PKI-based security certificates to initiate health information exchange.

References

1. Vagelis Hristidis, Peter J. Clarke, Nagarajan Prabakar, Yi Deng, Jeffrey A. White, Redmond P. Burke: A Flexible Approach For Electronic Medical Records Exchange. In: Proceedings of the international workshop on Healthcare information and knowledge management, Conference on Information and Knowledge Management, Arlington, Virginia, USA, Pages: 33 – 40. (2006)
2. Au, R.; Croll, P.: Consumer-Centric And Privacy-Preserving Identity Management For Distributed E-Health Systems. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences, vol., no., pp.234-234, 7-10. (2008)
3. Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, Yirong Xu: Hippocratic Databases. In: Proceedings of the 28th international conference on Very Large Data Bases, Hong Kong, China, Pages: 143 – 154.(2002)
4. Deepthi Rajeev, Catherine J Staes, R Scott Evans, Susan Mottice, Robert Rolfs, Matthew H Samore, Jon Whitney, Richard Kurzban, Stanley M Huff, 2010. Development Of An Electronic Public Health Case Report Using HL7 V2.5 To Meet Public Health Needs. In: The Journal of the American Medical Informatics Association, JAMIA; 17:34-41.

5. Song Han, Geoff Skinner, Vidyasagar Potdar, Elizabeth Chang: A Framework Of Authentication And Authorization For E-Health Services. In: Proceedings of the 3rd ACM workshop on Secure web services. (2006)
6. Janos L. Mathe, Sean Duncavage, Jan Werner, Bradley A. Malin, Akos Ledeczi, Janos Sztipanovits: Towards The Security And Privacy Analysis Of Patient Portals. ACM SIGBED Review, Volume 4 , Issue 2, Pages: 5 – 9. (2007)
7. Jing Jin, Gail-Joon Ahn, Hongxin Hu, Michael J. Covington, Xinwen Zhang: Patient-Centric Authorization Framework For Sharing Electronic Health Records. In Proceedings of the 14th ACM symposium on Access control models and technologies, Pages: 125-134. (2009)
8. Daglish, D.; Archer, N.: Electronic Personal Health Record Systems: A Brief Review of Privacy, Security, and Architectural Issues. In: World Congress on Privacy, Security, Trust and the Management of e-Business, 2009. CONGRESS '09, vol., no., pp.110-120, 25-27. (2009)
9. Sloane, Elliot; Leroy, Gondy; And Sheetz, Steven: An Integrated Social Actor and Service Oriented Architecture (SOA) Approach for Improved Electronic Health Record (EHR) Privacy and Confidentiality in the US National Healthcare Information Network (NHIN). In Americas Conference on Information Systems (AMCIS), AMCIS 2007 Proceedings. Paper 366. (2007)
10. Ajit Appari And M. Eric Johnson: Information Security and Privacy in Healthcare: Current State of Research. In: International Journal of Internet and Enterprise Management. (2009)
11. Vicky Liu, Lauren May, William Caelli, Peter Croll: Strengthening Legal Compliance For Privacy In Electronic Health Information Systems: A Review And Analysis. In: Electronic Journal of Health Informatics, Vol 3(1): e3. (2008)
12. Taylor, K.L.; O'keefe, C.M.; Colton, J.; Baxter, R.; Sparks, R.; Srinivasan, U.; Cameron, M.A.; Lefort, L.: A Service Oriented Architecture For A Health Research Data Network. In: Proceedings. 16th International Conference on Scientific and Statistical Database Management, vol., no., pp. 443- 444, 21-23. (2004)

Application Papers

Reified Literals: A Best Practice Candidate Design Pattern for Increased Expressivity in the Intelligence Community

Eric Peterson,

Global Infotek, 1920 Association Drive
Reston, VA 20191, USA
epeterson@globalinfotek.com

Abstract. Reifying literals clearly increases expressivity. But reified literals appear to waste memory, slow queries, and complicate graph-based models. We show where this practice can be comparable to unreified literals in these respects and we characterize the cost where it is not. We offer examples of how reification allows literals to participate in a variety of relations enabling a marked increase in expressivity. We begin with a case study in reified person names, and then extend this analysis to reified dates and simple reified scalar values. We show benefits for name matching and temporal analysis such as would be of interest to the Intelligence Community (IC). We then show how these same sorts of analyses can drive or inform any decision as to whether to reify literals.

Keywords: reified literal, semantic, ontology, expressivity, best practice, design pattern, Intelligence Community

1 Introduction

Reifying literals is not uncommon among popular ontologies and relational data models. But data architecture teams in the IC can draw from varied backgrounds and the use of reified literals may not be desired.

The practice may not be desired because it appears to waste memory, slow queries, and complicate graph-based models despite the increase in expressivity that it offers. We lay out simple, general metrics for judging such memory waste, slowness and complication. We show where the practice can be comparable in those respects to unreified literals and we characterize the modest cost where it is not. We show how reification allows literals to participate in a variety of relations which foster expressivity. Beginning with a case study comparing reified with unreified person names, we then extend the analysis to dates and simple scalar values. We then show how these same analyses can drive or inform any decision regarding whether to use or not use literal reification. This paper works toward establishing the reified literal design pattern as a best practice component.

We define reified literals as instances that represent literal values. A reification of a name string literal value might be an instance of type *Name* with a datatype property containing the value of the name string. This name instance might then be attached to a person instance by the *givenName* object property statement (See Figure 1).

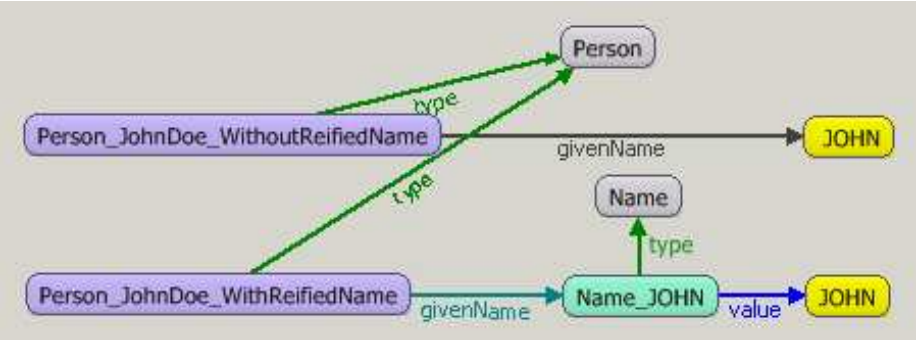


Figure 1: In the top unreified example, *givenName* is a datatype property statement. In the bottom reified example, *givenName* is an object property statement referencing a *Name* instance.

2 Current Practice, Related Work, and Contributions

Literal reification is not an uncommon practice. OpenCyc[1], Iode[2], and SUMO[3], ontologies have pervasively reified literals. DOLCE[4] leaves literal definition to extending ontologies.

W3C Semantic Web Best Practices and Deployment Working Group presented a draft by Hobbs and Pan[5] of a time ontology that uses only reified time literals. Project NeOn's Ontology Design Pattern Repository[6] contains a pattern for reified lexical items (terms). Many more main-stream examples exist.

The closest work related to the reified literals design pattern is Presutti and Gangemi's[7] content ontology design pattern requirements - to which reified literals comply[1].

Reified literals are not new. But we choose to characterize the virtues and cost of using this design pattern. We address some common misconceptions about this design pattern's performance by detailing memory footprint cost, speed, and design

[1] The reified literal design pattern is *computational* in that it is language independent and is encoded in a higher order language (OWL). It is clearly *small*. It is *autonomous* (deployable as a single file). It is *hierarchical* in that the class *Literal* must be subclassed for each particular literal. It is *inference-enabling* in that its reified instances become the foci of relationships that *say something* about the literal. Dates participate in Allen's interval calculus relations via transitive closure for example. The pattern is *cognitively relevant* in that it is intuitive, compact, and captures relevant notions in a domain. It is *linguistically relevant* in that we speak of names, dates, etc. as real things.

complexity. We give examples of the pattern's increase in expressivity. We show how to apply theses analyses to all literals.

3 Methods and Metrics

We describe simple, simple metrics to compare reified literal costs in memory and query speed with respect to those of unreified literals. With the relative complexity of reified literals and the benefits of their increased expressivity, however, we do not attempt to go beyond a qualitative description.

For memory usage comparison the two approaches differ structurally only by the type of one statement (after shared structure is amortized away). With the reified approach the type of the statement in question is *ObjectProperty* and with the un-reified approach it is of type *DatatypeProperty*. If datatype property statements are less compact in a particular implementation than object property statements, then the reified literal approach is correspondingly more compact for commonly referenced literals. The opposite is true if datatype properties are more compact in memory.

Our query speed comparison shows one type of reified literal queries that are faster than or equal to non-reified queries. This is due to the fact that there is a faster than or equal relationship between *(i)* an equijoin and *(ii)* a join equating two unreified literals. The other type of reified literal query is slowed by the speed associated with addition of a single equijoin. Further, we state that *(i)* an equijoin may be much faster than *(iii)* a join inexactly matching two unreified literals[2].

```
i:   {?person1 givenName ?reified_name .
      ?person2 givenName ?reified_name . }
ii:  {?person1 givenName ?unreified_name .
      ?person2 givenName ?unreified_name . }
iii: {?person1 givenName ?unreified_name1 .
      ?person2 givenName ?unreified_name2 .
      FILTER (likeTerm(?unreified_name1,
                       ?unreified_name2,
                       partialMatchSpec) }
```

Since the equijoin uses fast instance or integer comparison, it is faster or comparable to the join of two unreified literals[3].

[2] The *likeTerm* function is a non-standard extension to SPARQL for performing partial matching of strings. It is a more powerful version of the SQL *LIKE* keyword. The third argument is a partial match specification string that specifies the type of partial match and an optional matching template. In our implementation and others, inexact match is onerously slow compared to the speed of an equijoin. But the usage of a full text index, could make inexact search significantly faster.

[3] In RDF store implementations where strings are shared resources, the reified and unreified approaches are comparable because both are based on the matching of integers rather than strings.

The comparison of structural complexity we can base in part on statement counts and amortization of memory footprint comparison. But we ultimately rely on our reader's judgment as to the relative complexity.

The comparison of these sometimes negative costs against the benefits of increased expressivity is similarly subjective.

4 Case Study: Reified vs. Non-reified Names

First we treat the question of memory waste for reified names. Creating a new reified name each time a data source mentions that name would clearly waste memory. We, however, create a particular reified name instance just once for all its references to share. The cost of representing a non-reified name is one statement (<JohnDoe givenName "John">). The cost of a reified name would count the following object property statement (<JohnDoe givenName Name_JOHN>) plus the amortized cost of the shared reified name components for the name *John*: <Name_JOHN rdf:type Name> and <Name_JOHN value "John">. If just one reference to a person with the name "John" is in the data store, the extra reification cost is two statements – three times the non-reified approach. If the cost is shared between two references, the amortized extra cost is one statement; and among ten, the extra cost drops to one fifth of a statement (see **Figure 2**).

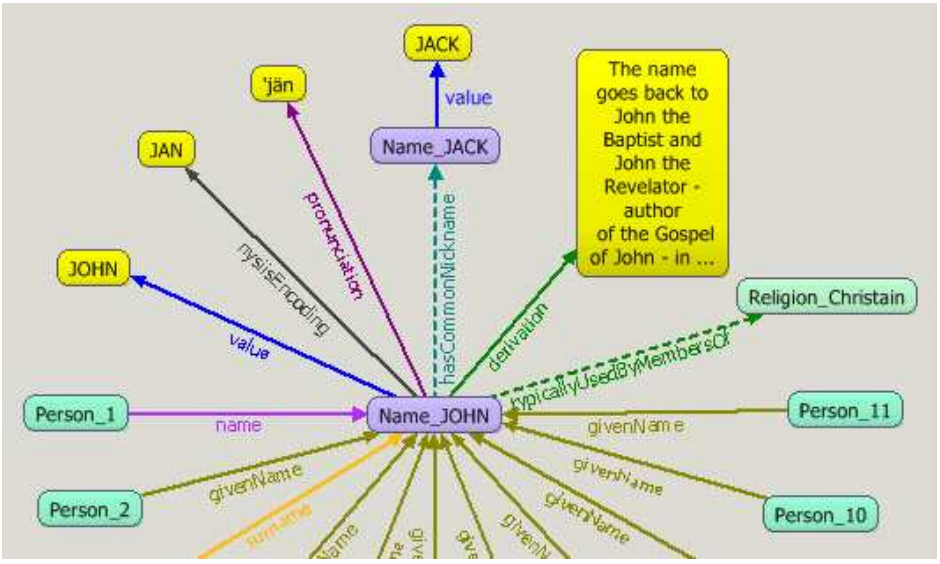


Figure 2: Because reified names are shared, the memory cost of the name *rdf:type* statement and name *value* statement is amortized out over all eleven name references. Note the several examples of the expressive power of reified names.

If names are common, the cost for their shared statements is negligible. So uncommon or *rare* reified names are individually costlier than unreified names, but the overall reified name cost is a function of the average number of references to the names.

Second, we treat the question of query speed. Exact name matching queries need not be string based. With name reification, one can match name instances rather than name strings. Implementations, then, can use integer comparison for speed equal to or faster than matching un-reified names.

When attempting an exact match query on a particular name, on the other hand, one must create the URI for that particular name, and one must create it in some repeatable canonical fashion. One must, for example, always translate the name *John* to precisely the same URI (e.g.: http://foo.gov/bar#Name_JOHN). This same name URI creation algorithm must be used for all names in the knowledge base. With these precautions, particular name matching also can also be a matter of simply comparing instances/integers rather than using an extra join to compare strings.

Inexact name matching requires an extra join when using reified names because the actual name string must be consulted. But this increase in query time is never large and is moot when using a system whose time for inexact matching overwhelms the cost of that extra join.

Next, we consider the structural complexity associated with reified names. The path length from a person node to her actual name value is one statement longer with the reified approach, but paying this price allows us to say things about names (see next section) in an organized fashion. We claim that if name meta-information is required, it is more intuitive to link it to the reified name instance and that the net effect is a reduction in complexity.

We now consider the benefits of reified names. Perhaps the most obvious benefit is being able to conveniently and intuitively say things about names and to reason about that information. A reified name can be linked directly to various information of interest to the IC such as its New York State Identification and Intelligence System (NYSIS) encoding, its variants, its nicknames, its ethnic derivation, a notion of its level of formality, its gender association, etc. (See **Figure 2**). Such information can be encoded in a system using non-reified literals, but the querier would need to understand the association between the name meta-information and the respective names. Clearly it is more intuitive to directly link the information about a name to some shared representation of the name. A newcomer need not know where the NYSIS information is stored. She simply queries on the name and sees, by inspection, that NYSIS information is associated with its corresponding names.

5 Generalizing Name Reification Results for Additional Literals

We discuss in detail the merits of reifying other literal types. We begin with dates, heights and weights.

As with reified names, reified dates are shared among all the events that reference them and, consequently, experience the same potential for memory cost amortization. Exact date match query speed, as for all reified literals, is at least comparable to the

non-reified case. Simple time range queries bounding the reified date's *xsd:date* value with two *xsd:date* values require an additional join Structural complexity of reified dates is clearly comparable to that of reified names. Expressivity-wise, dates are, of course, actual time intervals rather than simple time points. As an immediate result of date reification, one can concretely begin to better support temporal reasoning for IC applications. One can start by attaching beginning and end date-times to a date so as to allow dates to participate in precise time-interval-based queries. Reified dates can be unknown and yet known to be before some other known date. Such unknown date instances can participate readily in temporal interval overlap relations such as *temporallyContains* in order to bound the date if possible (see **Figure 3**).

With height and weight reification, amortized sharing cost efficiency is somewhat moot as likely usage tends toward few height and weight values[4]. Because even a minute difference in an exact match query is a miss, range queries are much more useful with scalars. Query speed in this case, therefore, is reduced by the cost of an equijoin, as actual weight and height values must be accessed. Reification increases expressivity such as being able to encode that one person is taller than another and both heights were unknown.

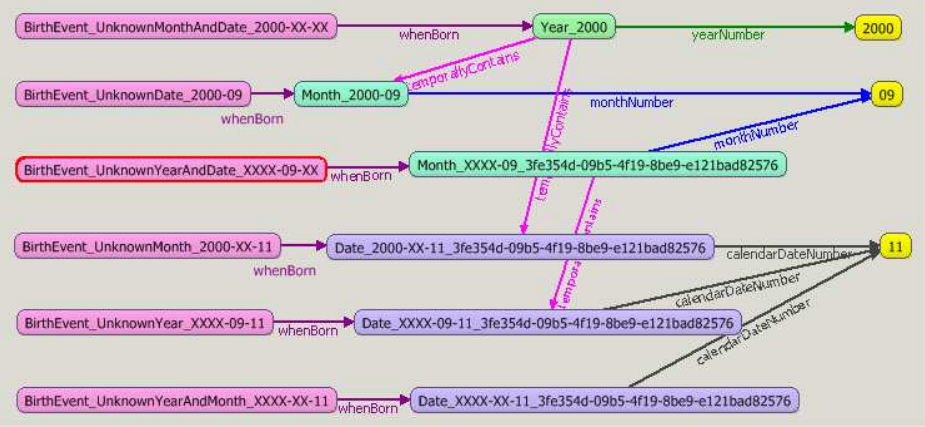


Figure 3: The following diagram shows six birth events each with a different sort of partial date. Years, months, dates, and birth events all have temporal extent and can, therefore, participate the various temporal interval algebra relations. Unknown date and months have URI names with embedded SHA 256 hash values to prevent them from coalescing with similar unknown dates and months. Were birthdays modeled as a datatype property rather than as reified dates, this sort of query-time expressivity would difficult and less intuitive.

These analyses apply to all twenty five of our twenty five literals. Without going into prohibited detail, all our literals are inherently sharable and, therefore, offer the same memory cost amortization potential. Zipf-Mandlebrot power law results are infrequently available for reifiable literals and they offer no guarantees that their exponents will be such that we can know that memory cost will be negligible[8]. All

[4] There are 289 English half inch values between zero and twelve feet inclusive.

but four of our date types and four of our scalars inherently lend themselves to using exact match and partial match, as with reified names. The reified name analysis, then, directly applies to these 17 literals. The two other scalar types behave just as height and weight with the range queries that are slower by one equijoin. The other date types are *Month, Year* and *TimePoint.* They are all simply date-like time intervals of various sizes, so their analyses are comparable to *Date*[5]

6 Results, Discussion, and Future Work

We established that common reified names have comparable in-memory cost to non-reified names. We similarly established that query speed for exact matching of reified names is equal or better than non reified names. And the additional speed cost of inexact matching is negligible in systems where inexact matching speed dominates that of a join. We argue that the overall structure of reified names and their metadata is simpler. We showed that reified names allow a sort of tightly linked expressivity that un-reified names do not.

We similarly analyzed dates, heights, and weights and found them to be slower by one join in interval queries. We found date expressivity to be significant and height and weight expressivity similar yet less likely to be justified by our data sets.

We distilled the following rules from the above analysis to help determine when the literal reification design pattern should be used:

1. Rare reified literals are individually costly, but the net cost is only a concern if there are very many rare types.
2. Range queries such as with reified scalars and dates are slower by an equijoin.
3. Inexact match queries over reified literals are slower by an equijoin. That equijoin is inconsequential on systems where inexact match dominates the query time.
4. Otherwise the speed and memory cost is comparable.

As we value expressivity, we found most of our literals to be reasonably strong candidates for reification. Our desire for expressivity also makes us less concerned as to how well amortized our shared structure is. We found all of our scalars to be weaker/marginal candidates because we have no present or near future need for scalar-related expressivity. Their inclusion would be based more on a desire to apply all design patters consistently.

In all cases, the value of the expressivity gain must be subjectively weighed against the possible cost in memory and speed. We have used literal reification for over fifteen years in the IC and in two different data integration projects *at scale*.

We expect that as we continue to observe the results of our choices to reify and not to reify literals, we will more finely characterize how to make such choices in the future. We expect to have opportunity to garner shared structure amortization statistics on our various reified literals.

[5] *TimePoint* is encoded as an interval as per common convention. *TimePoint* duration varies with the number of significant digits in the input.

7 Conclusions

Commonly referenced reified literals come at little or no significant cost in memory, speed, or complexity. Queries over such literals are never slower than the cost of one join with respect to unreified literals and are usually comparable. Where literal-related expressivity is specifically needed or expected, reified literals should be considered.

References

1. Cycorp Inc.: OpenCyc. http://opencyc.org
2. Hightfleet (formerly OntologyWorks): IODE, http://www.highfleet.com
3. Pease, A., Niles, I., and Li, J.: The suggested upper merged ontology: A large ontology for the semantic Web and its applications. In Proceedings of the AAAI-2002 Workshop on Ontologies and the Semantic Web, Edmonton, Alta., Canada (2002)
4. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., and Schneider, L.: DOLCE: A Descriptive Ontology for Linguistic and Cognitive Engineering. WonderWeb Project, Deliverable D17 v2.1 (2003)
5. Hobbs, J., Pan, F.: Time Ontology in OWL. Working draft, http://www.w3.org/TR/owl-time (2006)
6. Charlet, J., Vandenbussche, P.: Concept Terms. Ontology Design Patterns. http://ontologydesignpatterns.org/wiki/Submissions:ConceptTerms
7. Gangemi, A.: Ontology Design Patters for Semantic Web Content. ISWC 2005. LNCS, vol. 1729, pp. 262-276 (2005)
8. Zipf, G., Selected Studies of the Principle of Relative Frequency in Language. Harvard University Press, Cambridge, MA (1932)

Using New Standards to Develop IC Ontologies

Richard Lee

Booz Allen Hamilton
8283 Greensboro Drive
McLean, VA, 22102, USA
lee_richard@bah.com

Abstract: In this paper we describe recent work in adapting various new OWL and ontology standards to ontology development for the IC and DoD. We present work done to adapt the Universal Core Semantic Layer (UCore SL) standard ontology to support intelligence analysts. We show how new features in the OWL 2 standard can be used to make such ontologies simpler and more readable, and how they facilitate modeling the relationships of concepts across models. We present a proposed standard security model using OWL 2. We conclude with planned future ontology development using these standards.

Key words: Ontologies, OWL 2, Universal Core Semantic Layer, Standards

1. Introduction

Over the last several years, we have created OWL ontologies for use with the METS (Metadata Extraction and Tagging Service) system [1, 2], to represent the document metadata and semantic extraction results it produces. In the most recent iteration, these ontologies included and extended OWL versions of (parts of) SUMO, TWPDES, DDMS, ISM, code lists from ISO et al, and the "standard" Time and GML ontologies.

When the Universal Core (UCore) 2 standard [3] was released, it included a simple OWL taxonomy, so we added declarations to the master METS ontology to relate its concepts to those in the UCore taxonomy.

Barry Smith et al at NCOR started from the UCore model to develop a full foundational OWL ontology called the Universal Core Semantic Layer (UCore SL) [4]. In our recent (non-METS) work, we have developed an ontology based on it, to support a cell of IC/DoD analysts. We have also begun incorporating new OWL 2 [5] features.

2. Universal Core Semantic Layer Adaptation

In our most recent work, we were tasked with supporting a group of analysts by devising a consistent and inter-related set of models for their wide range of data sources and analytical processes, covering the usual assortment of people, organizations, and places, as well as numerous kinds of materials, equipment, and processes. We elected to create a set of ontologies, mapping to OWL each of:

- the schema for the desired subset of each data source (MIDB, TIDE, Artemis, …)
- the Palantir ontology we developed with the analysts
- the common organizational models called PMESII and CTAF

We also created a "master" OWL ontology, based on UCore SL, which covered all the concepts of interest to the analysts, and provided the OWL declarations needed to relate the concepts across all the other ontologies, for data mapping and correlation purposes.

In order to do this, we of course needed to extend UCore SL, adding whole sublattices of concepts under various of its concepts. For example, we have a handful of new classes refining UCore SL's *ActOfCommunication*. Similarly, we have new classes under its *Vehicle* and *Sensor*. In doing this, we borrowed heavily from SUMO [6]. For example, the whole area of *Equipment* / *Sensor* / *Vehicle* / *Weapon* is one where we found it expedient to insert a few higher-level concepts from SUMO. Since the various data sources, and UCore SL, differed on the question of which, if any, of the latter 3 concepts belonged under the former, SUMO's *Device* and some of its subclasses were the perfect root under which to organize and relate all those concepts from all the other models. Thus, for a representative sample of that part of the ontology, we have:

```
<owl:Class rdf:ID="Equipment">
  <rdfs:subClassOf rdf:resource="#Device"/>
  <owl:disjointWith rdf:resource="#ExplosiveDevice"/>
  <owl:disjointWith rdf:resource="#Sensor"/>
  <owl:disjointWith rdf:resource="#Vehicle"/>
  <owl:disjointWith rdf:resource="#Weapon"/>
  <rdfs:subClassOf rdf:resource="&art;Equipment"/>
</owl:Class>

<owl:Class rdf:ID="MeasuringDevice">
  <rdfs:subClassOf rdf:resource="#Device"/>
  <owl:disjointWith rdf:resource="#CommunicationDevice"/>
  <owl:disjointWith rdf:resource="#ExplosiveDevice"/>
  <owl:disjointWith rdf:resource="#Vehicle"/>
  <owl:disjointWith rdf:resource="#Weapon"/>
</owl:Class>

<owl:Class rdf:ID="Sensor">
  <rdfs:subClassOf rdf:resource="#MeasuringDevice"/>
  <owl:disjointWith rdf:resource="#Equipment"/>
  <owl:equivalentClass rdf:resource="&ucsl;Sensor"/>
  <owl:equivalentClass rdf:resource="&pal;Sensor"/>
</owl:Class>
```

```
<owl:Class rdf:ID="Vehicle">
  <rdfs:subClassOf rdf:resource="#Device"/>
  <owl:disjointWith rdf:resource="#CommunicationDevice"/>
  <owl:disjointWith rdf:resource="#Equipment"/>
  <owl:disjointWith rdf:resource="#ExplosiveDevice"/>
  <owl:disjointWith rdf:resource="#MeasuringDevice"/>
  <owl:equivalentClass rdf:resource="&ucsl;Vehicle"/>
  <owl:equivalentClass rdf:resource="&sumo;Vehicle"/>
  <owl:equivalentClass rdf:resource="&pal;Vehicle"/>
  <owl:equivalentClass rdf:resource="&avrs;Conveyance"/>
  <owl:equivalentClass rdf:resource="&tide;Vehicle"/>
  <rdfs:subClassOf rdf:resource="&meped;Equipment"/>
</owl:Class>

<owl:Class rdf:ID="Bomb">
  <rdfs:subClassOf rdf:resource="#Weapon"/>
  <rdfs:subClassOf rdf:resource="#ExplosiveDevice"/>
  <owl:equivalentClass rdf:resource="&sumo;Bomb"/>
  <owl:equivalentClass rdf:resource="&pal;Bomb"/>
</owl:Class>

<owl:Class rdf:about="&ucsl;Equipment">
  <rdfs:subClassOf rdf:resource="#Device"/>
</owl:Class>

<owl:Class rdf:about="&meped;Equipment">
  <rdfs:subClassOf rdf:resource="#Device"/>
</owl:Class>

… etc …
```

We also found it useful to borrow from SUMO to impose a bit more structure and detail in other areas, such as Geophysical and Geopolitical concepts.

3. OWL 2 Use for Simplifying Ontologies

The above examples follow the UCore SL practice of carefully declaring all the *disjointWith* relationships, including declaring each pair (redundantly) in both directions. One of the new features in OWL 2 is a pair of constructs for declaring this information in a cleaner, more compact fashion. Since some of the classes above are allowed to overlap (for example, *Weapon* can overlap both *ExplosiveDevice* and *Vehicle*), we don't have a nice clean partition which would enable removing all the *disjointWith*'s, but using the new *AllDisjointClasses* still helps somewhat:

```
<owl:AllDisjointClasses>
  <owl:members rdf:parseType="Collection">
    <owl:Class rdf:about="#Equipment"/>
    <owl:Class rdf:about="#ExplosiveDevice"/>
    <owl:Class rdf:about="#Sensor"/>
    <owl:Class rdf:about="#Vehicle"/>
  </owl:members>
</owl:AllDisjointClasses>
```

```
<owl:Class rdf:ID="Equipment">
  <rdfs:subClassOf rdf:resource="#Device"/>
  <owl:disjointWith rdf:resource="#Weapon"/>
  <rdfs:subClassOf rdf:resource="&art;Equipment"/>
</owl:Class>

<owl:Class rdf:ID="MeasuringDevice">
  <rdfs:subClassOf rdf:resource="#Device"/>
  <owl:disjointWith rdf:resource="#CommunicationDevice"/>
  <owl:disjointWith rdf:resource="#ExplosiveDevice"/>
  <owl:disjointWith rdf:resource="#Vehicle"/>
  <owl:disjointWith rdf:resource="#Weapon"/>
</owl:Class>

<owl:Class rdf:ID="Sensor">
  <rdfs:subClassOf rdf:resource="#MeasuringDevice"/>
  <owl:equivalentClass rdf:resource="&ucsl;Sensor"/>
  <owl:equivalentClass rdf:resource="&pal;Sensor"/>
</owl:Class>

<owl:Class rdf:ID="Vehicle">
  <rdfs:subClassOf rdf:resource="#Device"/>
  <owl:disjointWith rdf:resource="#CommunicationDevice"/>
  <owl:disjointWith rdf:resource="#MeasuringDevice"/>
  <owl:equivalentClass rdf:resource="&ucsl;Vehicle"/>
  <owl:equivalentClass rdf:resource="&sumo;Vehicle"/>
  <owl:equivalentClass rdf:resource="&pal;Vehicle"/>
  <owl:equivalentClass rdf:resource="&avrs;Conveyance"/>
  <owl:equivalentClass rdf:resource="&tide;Vehicle"/>
  <rdfs:subClassOf rdf:resource="&meped;Equipment"/>
</owl:Class>

… etc …
```

4. OWL 2 Use for Relating Ontologies

One of the principles in our modeling work was to represent all multi-faceted things as first-class objects, with classes in the ontology. In particular, it was clear that *Location*s should be represented in that way. By attaching properties to a *Location*, such as location containment (address contained in city contained in etc), location adjacency, location position (coordinates), even the Political, Military, Economic, etc circumstances of a location, the door is opened to reasoning about locations and the things at those locations.

Some of the RDB models we worked with made the same decisions on first-class objects, but many did not. For example, to relate a *Location* to some *Person*, *Organization*, *Event*, et al, the value of the relationship (*birthplace*, *residence*, *affiliation*, *destination*, et al) would often be, not a pointer to a *Location* record, but simply a string naming the location (often, just a country name).

Since one of our goals was to relate concepts across models, these string-vs-object differences were a problem. Again, OWL 2 introduces a handy construct which makes it possible to relate the two approaches. If, say, model **a** represents *birthCountry* as simply the name of a country, whereas model **b** represents *birthCountry* as a link to a country which has a name, we can indicate the equivalence via:

```
<rdf:Description rdf:about="&a;birthCountry">
  <owl:propertyChainAxiom rdf:parseType="Collection">
    <owl:ObjectProperty rdf:about="&b;birthCountry"/>
    <owl:DatatypeProperty rdf:about="&b;name"/>
  </owl:propertyChainAxiom>
</rdf:Description>
```

5. OWL 2 Use for a Standard Security Model

When the new OWL 2 model was discussed at the 2008 Semantic Technology Conference, it was noted that the new annotation property capabilities were suited for capturing information such as security, provenance, and confidence, all uses of great interest to this community. We have accordingly mapped the IC's recently-released XML security model, IC-ISM v3, into an OWL ontology called ISM3 using the new constructs.

We have defined a property for each of the ISM v3 XML attributes, a *Security* class as their domain, and a *security* annotation property to relate a *Security* class instance to anything. We have mapped each of the "CVEs" (Controlled Vocabulary Enumerations) defined by the IC-ISM v3 XML specification into the OWL equivalent. For example:

```
<owl:Class rdf:ID="CVE_Classification_US">
  <rdfs:label>CVE: Classification (US)</rdfs:label>
  <rdfs:comment>allowed values for a classification, US-
only</rdfs:comment>
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#U">
      <rdfs:comment>UNCLASSIFIED</rdfs:comment>
      <ism:security rdf:resource="#U-USA"/>
    </owl:Thing>
    <owl:Thing rdf:about="#C">
      <rdfs:comment>CONFIDENTIAL</rdfs:comment>
      <ism:security rdf:resource="#U-USA"/>
    </owl:Thing>
    <owl:Thing rdf:about="#S">
      <rdfs:comment>SECRET</rdfs:comment>
      <ism:security rdf:resource="#U-USA"/>
    </owl:Thing>
    <owl:Thing rdf:about="#TS">
      <rdfs:comment>TOP SECRET</rdfs:comment>
      <ism:security rdf:resource="#U-USA"/>
    </owl:Thing>
  </owl:oneOf>
</owl:Class>

<ism:Security rdf:ID="U-USA">
  <ism:classification rdf:resource="#U"/>
```

```
    <ism:ownerProducer rdf:resource="#USA"/>
  </ism:Security>
```

In contrast to the usage of *Security* above, which annotates each entry in the enumeration with its security markings, we note that usual practice would be the use of annotated axioms, each of which simultaneously asserts and annotates a triple:

```
  <owl:Axiom>
    <owl:annotatedSource rdf:resource="#ID1"/>
    <owl:annotatedProperty
rdf:resource="http://example.com/example.owl#memberOf"/>
    <owl:annotatedTarget rdf:resource="#ID2"/>
    <ism:security rdf:resource="#Sec1"/>
  </owl:Axiom>
```

We should note that ICS500-21 "Tagging of Intelligence and Intelligence-Related Information" directs that all XML documents shall use the ISM XML standard for security markings. This is of course impossible for XML languages such as RDF/XML. But the rationale for that directive is obvious, and applies to OWL data as well. We urge the community to agree on a standard OWL ontology for security, so that it can be approved as an alternative, and provide the same benefits for OWL use that agreeing on ISM XML does for XML use. We offer this as a possible approach for that standard. We suggest that a similar standard for provenance (sourcing) would be beneficial as well.

6. Future Work

We plan to:

- incorporate mappings to UCore SL into the METS ontology
- return to the other project to model and map additional data sources and concepts
- continue retrofitting OWL 2 constructs in both
- continue devising ontologies such as IC-ISM v3, ideally in coordination with others across the community

7. References

1. Lee, R: The Use of Ontologies to Support Intelligence Analysis, Ontologies in the Intelligence Community Conference (2007)
2. METS: http://purl.org/mets
3. Universal Core: https://www.ucore.gov/
4. Smith, B., Vizenor, L., Schoening, J.: Universal Core Semantic Layer, Ontologies in the Intelligence Community Conference (2007)
5. OWL2: http://www.w3.org/TR/owl2-overview/
6. SUMO: http://www.ontologyportal.org/

Semantic Wiki for Visualization of Social Media Analysis

Daniel Reininger, David Ihrie, and Bob Bullard

Semandex Networks Inc., 5 Independence Way, Suite 309,
Princeton, NJ 08540 (609) 681-5382
{djr, dihrie, bob}@semandex.net

Abstract. A semantic wiki provides visualization of social media analysis applicable to military Information Operations and law enforcement counter-terrorism efforts. Using inputs from disparate data sets, semantic software exports data to link analysis, geospatial displays, and temporal representation. Challenges encountered in software development include the balance between automated and human assisted entity extraction, interoperability with existing visualization systems and ontology management.

1 Introduction

Social media analysis is an important part of military and law enforcement operations [1] [2]. The analysis requires the ability to model and extract significance from the social media interactions of persons and organizations of interest. This analysis must be done in real time and in the virtual, collaborative workspaces of the law enforcement and intelligence communities.

This paper outlines issues identified during the development and demonstration of a software tool to provide shared visualization for social media analysis in selected government environments. We developed and tested a software application pursuant to a federally sponsored program titled Information Networking for Operational Reporting and Monitoring (INFORM). The project was designed to facilitate collaborative analysis and workflows for elements of the U.S. Marine Corps, the U.S. Special Operations Command, and the U.S. Department of State.

Existing information sharing applications available to the user community included the Combined Information Network Data Exchange (CIDNE) [3], Intellipedia [4], and the Net-Centric Diplomacy portal [5]. Each of these programs provided an avenue for information sharing and multi-agency collaboration, primarily by making documents—whether finished reports or community-updated web pages—available to a broad community. However, each of these systems exhibited a common disadvantage that the INFORM program was designed to help mitigate: tactical users needed to model information of local interest that could not be easily captured in CIDNE, NCD, or Intellipedia in a way that facilitated efficient and dynamic query, retrieval and display. A solution had to provide three advantages over the existing systems. First, the solution had to provide the user with a means to rapidly tailor the

information model to handle novel concepts encountered at the lowest tactical echelons. Second, the solution had to allow for the dynamic assembly of documents so that views of information were automatically and continually updated throughout the knowledge base; new social links had to be instantly recognized and published as soon as these links were discovered by the system. Third, the solution had to provide a means of efficient manual and automated query and display, including the ability to export data extracts to specific visualization applications (external to this software solution) designated by the user community.

The goal of the INFORM program was to create a web-based application with these capabilities that supported Information Operations. The technical approach was to develop a semantic wiki for data capture, analysis, and display. The desired end state was the ability to link entities contained in reports, open source articles and other sources encountered by users, creating a semantic graph that helped with social media analysis rather than simply serve as a document management system. A semantic approach met the end state requirements and offered additional advantages. First, data could be combined from disparate sources. Some data were highly structured and amenable to computer processing, while other data were unstructured, with syntactic incompatibilities that inhibited automated data ingestion to the system. We used a semantic schema and domain-specific ontology to parse information, generate concept instances, and represent relationships identified in the data. Second, a web-based wiki provided distributed access and rapid dissemination of information for multi-user collaboration. It provided a platform that generated and transmitted alerts based on changes in collective knowledge, such as the discovery of additional relevant information. Individual users set their own alert parameters and received individual notifications, by email or web-based chat, that included embedded links for one-click viewing of updated information.

2 Discussion

The use case in which the software was applied involved social media analysis supporting psychological operations. Specifically, we used the new semantic application to perform analysis of a selected target audience in accordance with existing doctrinal procedures [6]. This involved the review of data from open source media, combined with data from additional sources, to support an assessment of social groups, subgroups, and individuals within a population.

Visualization of the results of social media analysis is essential to effective target characterization for influence operations. The analysis in this project supported the initial study of a subject audience and evaluated measures of effectiveness to determine behavioral change, as evidenced in differences in social media behavior. Variations were observed both in social media content generation and activity patterns. The approach taken to determine changes in social media behavior was driven by the data available, which was a function of available sensors and information access.

2.1 Input Interfaces

The semantic software developed supported interfaces with databases, emails, RSS feeds, web pages, and spreadsheet files customized to support existing concepts of operation. An issue we faced was achieving the optimal balance between automated and human-assisted data ingestion. Uploading spreadsheets is a simple means of automated input; however, extracting pertinent information and context from unstructured text is also an important component of social media analysis, since statistical display of themes extracted from social media (e.g., blogs) are an indicator of social sentiment. Research comparing human-assisted entity extraction from text with automated methods, in efforts to enable automated network node/edge determination, indicate that the methods are complementary [7]. A mix of human involvement and automated processes provides the ideal balance of speed, ease, and validity. We used automated entity recognition in text coupled with human-validated associations to input data into a semantic graph. Additionally, for statistics not requiring additional human validation (such as summations of statistical analysis of sentiment), we incorporated automated data ingestion and automated visualization.

A customized data loader feature was developed for the software to facilitate automated upload of information. The primary challenge encountered during automated input from source databases was gaining an understanding of the structure of source data without a descriptive model of that data. This understanding was integral to writing the appropriate SQL statements to retrieve data in the desired format. This obstacle was overcome by manual inspection of columns in the source database, looking for promising column names, and examining the contents to verify that the mapping was appropriate. This mapping and tailored ingestion [detailed in reference 8] was critical to harmonize geospatial and temporal data from disparate datasets. This process would have been greatly aided by a mechanism to find similar names, similar content, and matching enumerations in order to help understand how the source data mapped to the target, as well as a mechanism to build and test the necessary SQL statements that ultimately retrieved data from the source. Once complete, the data loader mapped data into the semantic database while ensuring that incoming data met certain standards. The data loader template built and loaded concept instances and properties, and then built relationships between the concepts to form the semantic graph.

2.2 Output Interfaces

Visualization was delivered in three distinct categories: link analysis, geospatial representation and temporal representation. Rather than duplicate efforts to develop capabilities that would require user training, we focused on utilities for export of relevant data to existing third party applications already commonly employed by the user community.

Figure 1 shows the system's architecture and input/output interfaces. Input interfaces included databases, emails, RSS feeds, web pages, spreadsheet files. External applications included, but were not limited to, link analysis, and geospatial visualization of select data. Figure 2 shows the output displays of results into Link

Analysis tools and geospatial visualization. Results summarized in the person's page can be visualized as a link chart and in geospatial representations.

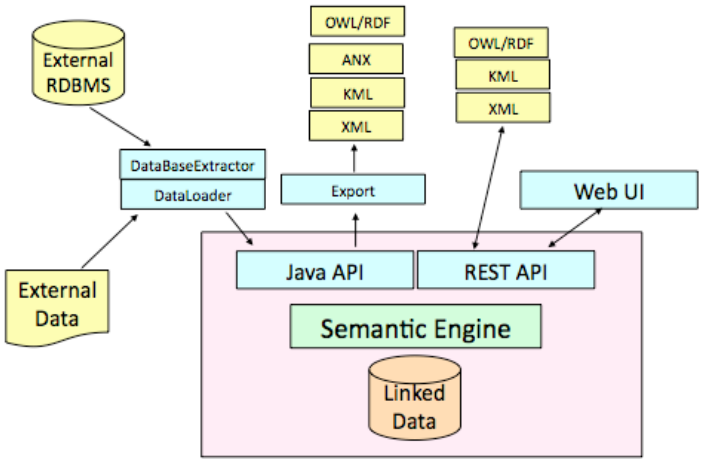


Fig. 1. Semantic wiki software architecture.

Fig. 2. Export to external applications.

Figure 3 shows a geographic display, using Google Maps, of population sentiment derived using statistical aggregation of data related to individuals, exported as KML. Geographic clustering and display of sentiment statistics derived from social surveys is an accepted methodology within the military for obtaining "ground truth" [9].

Temporal views complemented geographic displays. Software views based on adjusting time frames can indicate periods of high and low centrality, productivity,

and information dissemination; however, contextual cues that compliment temporal views are critical to gaining a true understanding of social interactions [10]. We developed a tailored display to fit the unique requirements of temporal visualization for social communications between individuals, which we could not obtain using existing external applications.

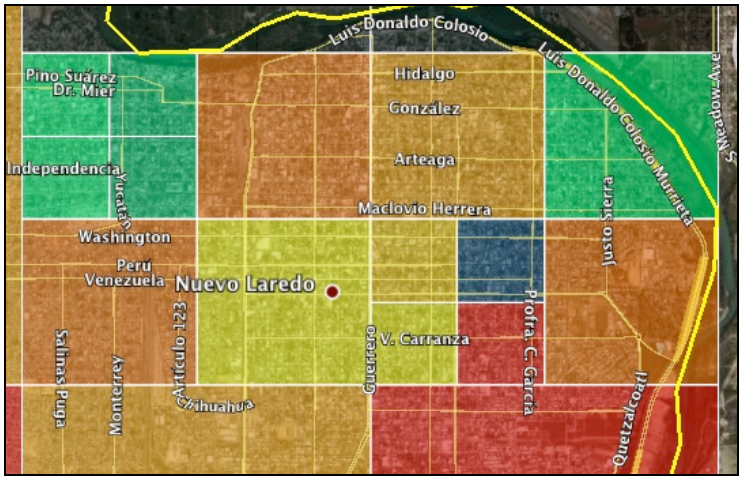


Fig. 3. Geographic display of population group sentiment using Google Maps.

Tailored representations included a heat map that showed activity by time and day of week to identify changes in individual social communications behavior. Filters provided adjustable date ranges and the ability to select the type or types of interaction (phone call, text message, etc.) displayed by the software.

Such visualization of the results of social media monitoring and analysis offers direct application to addressing the challenges and opportunities that result from the widespread use of social media, and its necessary inclusion in an environment of Information Operations. The utility of these visualizations applies equally to law enforcement, particularly in a counter-terrorism role.

During the course of this project, we encountered several salient issues that merit further research to expand the capabilities for social media analysis and visualization. We next present some possible approaches to ontology management, but leave the recommendations as open-ended avenues for the development of the field.

2.3 Ontology Management

This project developed a common schema for representation of information of interest to multiple potential user communities, including psychological operations, civil affairs, and intelligence information related to people, regions, countries, events, threats, and similar topics. This common schema provided the foundation for semantic information modeling that resulted in the ability of users to contribute to and draw on a common information picture expressed in a semantic graph.

There has been discussion of implementing a high-level, domain-independent ontology to provide a framework from which disparate systems in the government and military arenas could derive domain-specific ontologies [11]. Lacking such a foundational ontology of universal application, we developed a semantic schema and a domain-specific ontology for this project. Our solution did not need to provide formal inferencing; accordingly, our application did not require a formal ontology. We did need enough structure to capture associations in the data to present, for example, the optimal path to get a message to influential individuals within a community of interest.

Certainly, the integration of social media analysis software with existing enterprise information systems requires either ontological commonality or ontological bridging to enable effective interface across domains. Such a bridging ontology [12] might be useful to more easily ingest additional data sets into the semantic graph, or export data from the semantic graph to other databases without the need for customizing data loader templates or data export functions.

Even within a single domain and system, users required a method of tailoring the ontology by extending it "on the fly" to accommodate new concepts. For example, while examining media inputs, a user identified the need to add a "Tweet" as a subtype of instant message. Fortunately, the user interface provided a means by which the user could, without the need for software programming, spontaneously add relevant concepts and integrate this data into the analysis picture without corrupting the structure of the database or impacting functionality of the software. While the ontology remained "informal" in that users could extend it, the software-enforced adherence to that ontology "formalized" its acceptance by the user community.

The requirement for an adaptable ontology poses conflicting challenges. First, a user faced with a new classification of information must be able to define the new entity in the ontology. Second, this process must be managed collaboratively. If every user is continually modifying the structure of the ontology, it will rapidly cease to function; data calls for visualization will fail. Instead, designated stewards of the ontology in a user organization must make necessary modifications without recourse to technical support. Developers must take the requirements for agile schema into account in the planning and design phases of the software design life cycle to ensure that users can keep the application relevant in the ever-evolving conditions of social media analysis.

Two lessons learned, and successfully applied, were that recognition engines and recommendation engines can assist with ontology management. A recognition engine was incorporated that functions on free text input from files or websites, and preprocesses the text before presenting it to the user's view. It highlights entities in the text that are already known to the semantic wiki, such as the name of a specific person or place. The engine then uses a semi-automated process to help the user delineate relationships between existing entities and new entities created by the user. This human-to-machine interface prevents the user from unnecessarily creating new entities, and prevents the software from making errors of association that are a common byproduct of fully automated text recognition and database ingestion.

A recommendation engine provides assistance if the operator is still not satisfied with existing subtypes (as in the above example of the "Tweet"). First, the user interface allows the addition of a new subtype to the ontology. The software then

repopulates the modification to all user displays, allowing collaborative awareness and use of the new subtype. Now a recommendation engine can provide the user with awareness of alternatives, including newly created subtypes; when the software processes incoming text, this recommendation engine presents the user with a list of available entity types with which to tag new entities. "Tweet" is now recommended as an option that the user can select. This utility optimizes visibility of the ontology and limits the unnecessary duplication of subtypes.

2.4 Results

A source corpus of data composed of 204 files in four formats (.jpg, .html, .txt, .doc) produced a semantic graph of 4310 concept instances in a semantic database totaling 102 MB. This represented the interrelations of 585 events and 196 persons.

The domain-specific ontology expanded from eight basic concepts (Person, Organization, Place, Event, File, Characteristic[1] and Watchlist[2]) to 262 types of pages defined by use case analysis and by direct user additions. For example, "Communication Event" is a type of "Event" and "Tweet" is a type of "Communication Event". Also a "Facility" is a type of "Place" and a "Broadcasting Station" is a type of "Facility". However, actual data modeling for social media analysis during the practical application phase of this program utilized only 63 types of pages, or 24% of the total available. This suggests that users will, even when presented with a myriad of choices in modeling data, often use commonly recognized entity types. It also evidences the effectiveness of recognition and recommendation engines in limiting the inclination of users to modify an adequately developed ontology.

3 Conclusion

This project has resulted in the development of software that provides social media analysis and visualization for specific customers in the government community. We have developed and tailored a commercially available semantic software solution to integrate disparate social media data sources using automated and machine-assisted techniques that promote data validity and collaborative accessibility. While the results apply directly to military Information Operations and law enforcement counter-terrorism efforts, we believe that the issues faced are widely applicable to researchers, software developers, and program managers in other domains related to the semantic exploitation of social media.

An issue of interest to the reader community is the delicate task of finding the balance between automated and machine-assisted (human-validated) data ingestion. This is a balance that all data analysis applications must attain to preserve data

[1] Characteristics model distinctive features of any entity (e.g., person, thing, event, place). For example, "tall", "long hair" and "caucacian" can be person's characteristics.

[2] A Watchlist page has two links: (Has Member) *Page* and (Is Watchlist Of) *User*. When a member page is updated, the user will be notified of the update.

validity, without sacrificing scalability. Input methods such as spreadsheet and unstructured text ingestion promote speed and utility, while entity recognition and user-supervised text exploitation validate input to a collective database.

We have leveraged existing third party applications preferred by the user community to visualize information, including relations between abstract concepts in social media, using link analysis and geographic display. Additionally, selected user requirements have been met by the development of tailored temporal displays.

Ongoing challenges include ontology management, where the requirement is to provide the user with a mechanism to continually refine a domain-based ontology. While adaptable software, with recognition and recommendation engines, allows for a user-extensible ontology, the broader issue of cross-domain mapping using a bridging ontology offers opportunity for further study.

References

1. U.S. Department of Defense (USDoD) Joint Publication 2-01.3: Joint Intelligence Preparation of the Operational Environment, pg. xii. Department of Defense, Washington, DC (2009)
2. International Association of Law Enforcement Intelligence Analysts (IALEIA). 2004 Law enforcement analytic standards, published in association with the U.S. Department of Justice (2004)
3. Intelligent Software Solutions, Inc.: CIDNE, http://www.issinc.com/solutions/cidne.html
4. Wikipedia: Intellipedia, http://en.wikipedia.org/wiki/Intellipedia
5. Pack, D.: Profiling and testing procedures for a net-centric data provider. SPAWAR System Center Charleston, North Charleston, SC (2005)
6. U.S. Department of the Army (USDoA). FM 3-05.302: Tactical psychological operations tactics, techniques, and procedures, pp. 6-4 to 6-11. Headquarters, Department of the Army, Washington, DC (2005)
7. Graham, J., Carley, K., Cukor, D.: Intelligence database creation & analysis: network-based text analysis versus human cognition. In: Proceedings of the 41st Hawaii International Conference on System Sciences (2008)
8. Semandex Networks, Inc: Rapid semantic integration of data using the Tango DataLoader framework, http://www.semandex.net/servlet/DownloadServlet?id=397
9. U.S. DoA 2005: FM 3-05.302, pp. B1-B12
10. Gloor, P., Laubacher, R., Zhao, Y., Dynes, S.: Temporal visualization and analysis of social networks. In: Proceedings of the North American Association for Computational, Social and Organizational Science Conference (2004)
11. Semy, S., Pulvermacher, M., and Obrst, L.: Toward the use of an upper ontology for U.S. Government and U.S. military domains: an evaluation. MITRE Corporation, Bedford, MA (2004)
12. Gilson, O., Silva, N., Grant, P.W., Chen, M.: From web data to visualization via ontology mapping. Computer Graphics Forum, 27, no. 3 (2008)

Collected Imagery Ontology: Semantics for Airborne Video

Alexander Mirzaoff

Geospatial Systems
ITT
Rochester, New York. August 2010
585-269-5700
Alexander.Mirzaoff@ITT.com

Abstract. A prototype Video Imagery Ontology has been developed to derive video imagery intelligence, VideoIMINT. The ontology includes the development of classes and properties to address video image content, and video collection metadata related to platforms, sensors and collection operations. Preliminary feature extraction of video imagery content classes was functionally utilized to identify important video segments in an integrated viewer. Integrated data storage systems and fusion processes are proposed and discussed.

Keywords: Ontology, semantic, imagery, video, intelligence.

1 Introduction

For decades, the increasing volume of imagery data has been a growing challenge for the military and intelligence communities, "*too much to look at…*" and "*most of the bits end up on the floor*". The coming of age of Video Intelligence Surveillance and Reconnaissance (VISR) has only exacerbated the problem by orders of magnitude. For areal coverage with multiple, high resolution cameras [1], operating at two hertz and greater frame rates, data volume is now calculated in yotta-bytes (10^24 bytes). Notwithstanding the computational, storage and networking problems associated with this amount of data, finding content via database searches through these many instances of data becomes very problematic. Lt. Gen. David A. Deptula remarked that the Air Force could soon be "swimming in sensors and drowning in data." [2]. The recognition in this comment of the sensor, as well as the data volumes, as part of the overwhelming information glut, is very important and telling as to how these systems are utilized.

Ontology structures, as a filter for domain information, and ontology enabled rules of organization, present many advantages to help navigate and automatically use such volumes of information. Ontologies can address apparent substantive conflicts of

detection when confronted with phenomena represented by different sensors (panchromatic, multispectral, infrared, RADAR…) on various platforms, collected under widely different circumstance in an automated, sensor to computer to human workflow.

Collected imagery data, and to a larger extent, the information represented, is an organizational, if not a metaphysical, challenge. Consider just two sensors, infrared (IR) and RADAR on the same aircraft. Does all the IR data go here to the IR data bin and all the RADAR data over there in the RADAR bin? Suppose we have both from the same area on different days, or perhaps one for 5 minutes and another data set for 5 hours? Specifically, how are such diverse collections correlated? Do we organize by spectra, by location, by time, or perhaps platform? Is intelligence driven or prioritized by location, time, content, or all these attributes and more? Obviously, these elements are all important, while to complicate matters, the importance varies from mission to mission.

Additionally, there are operational classes that impact domain organization; including aspects of, surveillance utility or operational reconnaissance. Elements of platform specification and platform performance, sensors and sensor performance, and products derived from mission data are also important. The ontological effort is to separate these concepts so that sensor performance, for instance, can be applied to any mission, describing sensor success in some qualitative and quantitative manner.

However, the most differentiating property of intelligence collections is data content: data defined features and objects extractable from a particular collection. While all other elements, or classes, of imagery collection, such as which aircraft, which sensor, provide a rich compilation of schematic information – subclasses and properties – it is the semantics of imagery content that moves this structure from the utility of databases to the world of ontologies. To understand this difference, consider the query "which *sensor* observed the IED explosion at *location x* during *time t*", as compared to, "were *individuals* observed prior to IED explosion at *location x* during *time t*". While building a database schema construct for object concept *sensor* is non-trivial, adding a class such as *individuals* which is, in fact, detected content of imagery, becomes a significant semantic encounter.

Thus, the initial effort has been to define, organize and build an ontology of the VISR domain, including imagery content classes, to enable automated data processing and domain query and management. Subsequent efforts will use this structure to develop the complex logic and relationships of this domain. Flexibility and change are driving principles so that the resulting ontology can be edited: modified as new knowledge is gained, particularly as imagery context is developed with more and more elements extracted from imagery data.

2 Initial Classes

The VISR classes that were initially proposed include the following:

Domain Classes	Subclasses
1. ***Platform Sensor and Sensor Operation***	a. Platforms b. Sensors c. Operational Parameters d. Calibration and Quality Metrics
2. ***Collection and Collection Performance***	a. Collection Variables b. Collection Operational Parameters c. Collection Performance Metrics
3. ***Mission and Targets***	a. Mission Description b. Detection and Characterization
4. ***Imagery and Exploitation Products***	a. VideoIMINTHierarchy b. Product Descriptions c. Product Utility d. Data Assurance Metrics
5. ***Integrated Ontology***	a. Relationship and Rule Algorithms

Table 1. Initial Organizational Construct

Due to programmatic limitation, only classes 1, 2, and 4d were developed. The ***Integrated Ontology*** concept was dropped because the major classes covered the domain rather completely for this application, requiring no further integration, and relationships were an outcome of structure, even at the lowest levels of, for example, sensor calibration and product utility.

The AAF Profile for Aerial Surveillance and Photogrammetry Applications (ASPA) specification [3] provides an excellent starting structure to begin differentiating such concepts as performance and metrics in this ontology. This metadata specification is an XML type structured document that lends itself well to transition into a Resource Description Framework (RDF) for use in a hierarchical ontology.

The ASPA specification covers a great deal of video support information, including where and when it was collected as well as sensor data and platform data, so that the consequent instances of a particular mission, reflected in the video metadata, easily populate the ontology classes of platform and sensor. Such information, semantically consistent, and further constrained by the ontology structure, can form the basis for subsequent queries that reveal much richer content than at first apparent. In fact, structuring the ontology in this manner sets up the entire domain in a logical and computationally complete structure. To further enhance the subsequent utility, the ontology is written in the Ontology Web Language (OWL) standardized by the World Wide Web Consortium (W3C) in the Descriptive Logic (DL) version.

The VISR ontology design was based upon an upper level ontology utilized by the National Center for Ontology Research (NCOR). In this approach, Entities and Events constitute the two main component classes of the upper level, with an entity comprised of two main branches, the Dependent and Independent Continuants.

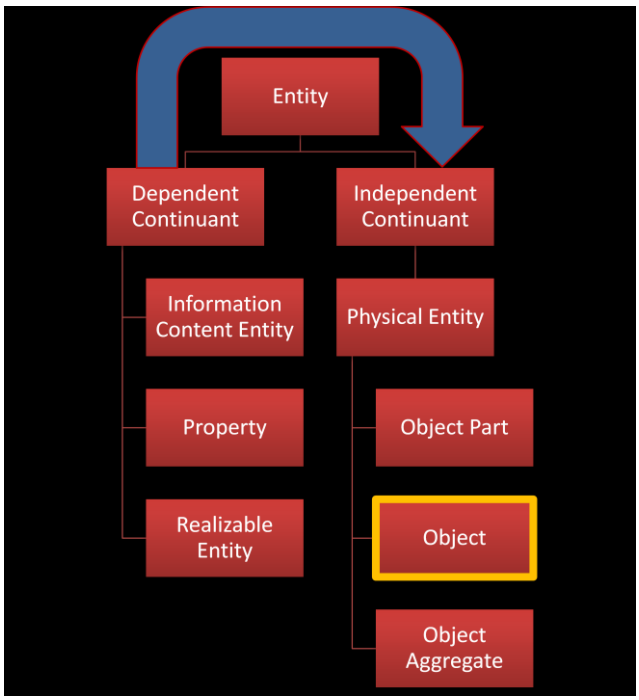


Figure 1. Upper Classes of the NCOR ontology.

Working down through *Independent Continuant* branch to the class of *Object*, we find that this area of the ontology includes subclasses for *Information Bearing Entity*, *Image Bearing Entity*, and both *VideoImage Bearing Entity* and *StillImage Bearing Entity*. Including these as subclasses of *Information Bearing Entity* allows for the later expansion of the class to include other sensor data such as from SIGINT or MASINT collection systems.

On the *Dependent Continuant* side of the ontology structure, we find the *Information Content Entity* from which is derived a *Descriptive Information Content Entity*, and subsequently the class *Image* and a subclass *Video Image*, an image that contains a moving (or extended temporal) representation of some Entity or Event, or *Still Image*, an image that contains a non-moving (or limited temporal) representation of some Entity or Event. These classes are what we would normally think of as the image or the video, while the *Image Bearing Entity*, including both *VideoImage Bearing Entity* and *StillImage Bearing Entity* are bearers of some *Video Image* or some *Still Image* found in the *Dependent Continuant* side of the ontology.

This differentiation provides for the description and definition of additional object classes such as *Pixel* and *Geospatial Region* as an *Independent Continuant* of the pictures that may be subsequently created. Additionally, the ontology can describe classes of *Object* such as *Facility*, *Vehicle* and *Sensor* independently of any particular *Facility*, *Vehicle*, and *Sensor*, again providing a means to specify facilities that are then imaged with particular attribute subclasses such as *Airport* or *Aircraft*.
There is another type of *Physical Entity* class called *Object Aggregate*, of which a subclass is a *Platform*. This *Platform* has properties denoted as `has_part,` such as **ImageSensor** and another `has_part`, **Aircraft**. In this manner, we can now construct a complex object, a **UAV**, as shown in Figure 2.
So with such a construct, we have the ability to present an image, describe its content (through some content extraction algorithm, such as feature extraction or automated target recognition) and relate that content to associated collection parameters (e.g. sensor, frame location, time, altitude…) as well as quality metrics of sensor performance that would be reflected in pixel characterizations, for example.

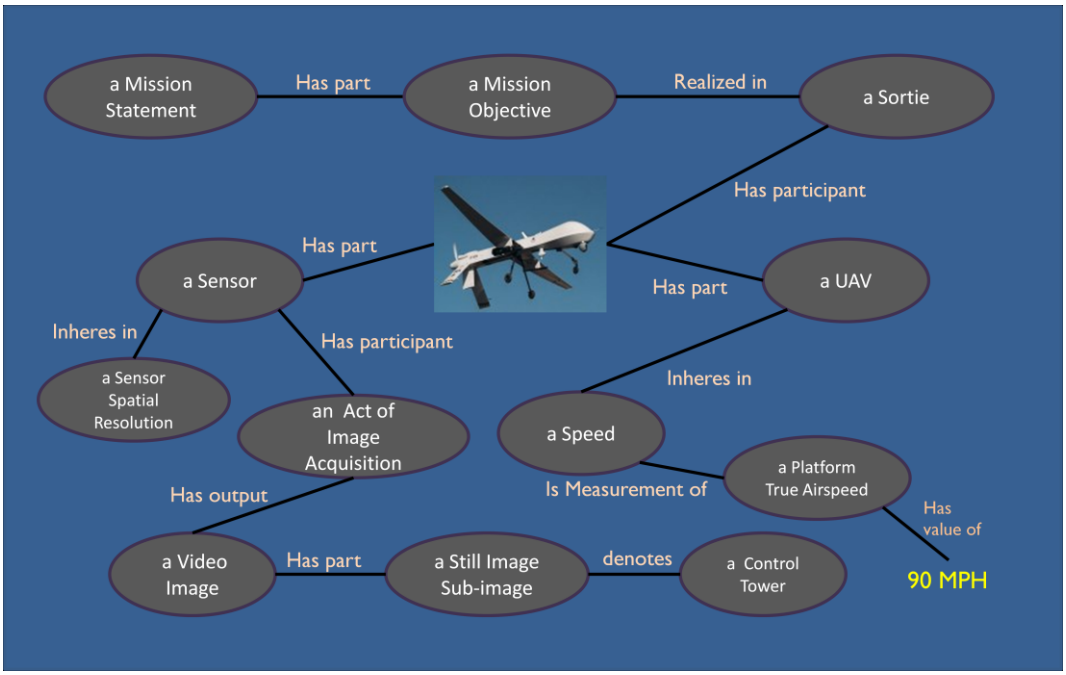


Figure 2. Real World Object as multi-class constructs

Note also that the instances of aircraft properties such as Speed, Direction and Location can be found in the ASPA metadata that is passed with the Predator UAV Datalink Local Metadata Set data elements (*i.e.* video metadata) [4]. Furthermore, since this information is dynamic, it can be updated and associated with any frame of the video collection.

3 Data as Image Content

A key aspect of making this VideoIMINT ontology useful is the ability to extract content from image data. That is, to be able to identify objects (e.g. vehicles, people, weapons), preferably in an automatic manner, from the collected data. There are two aspects to this problem: first, the image content itself – the targets of interest, and second, the support data provided by the sensor and sensor platform as well as from other opportunistic sources. First, we will review the challenges associated with discovering imagery content.

Ontological classes of content at first appear to be straight forward – vehicles, facilities, infrastructure, people… yet extracting these target object instances to populate these classes is a complex and elusive process to undertake in an automated manner. Manual tagging is an option that will be used for the foreseeable future, and facilitating this functionality in an efficient, icon driven manner is an additional objective of the VideoIMINT ontology effort, as is editing the classes of the ontology to be able to add additional target classes.

Automated feature –object extraction from imagery, and in this case video, continues to be an evolving and complex process. Much of the early efforts in understanding and classifying data from overhead remote sensors were in the area of Geographical (or more recently, Geospatial) Information Systems (GIS). For earth observing systems, in order to classify sensor data and build an ontology, Camara *et al* [5] originally argued for a concept of objects as a subset of geospatial fields while acknowledging the overly generic boundaries of this idea. With this approach, everything in the world is a field or an object in the field. This bodes well for constructing a subsequent ontological model since the separation of objects is axiomatic. The problem with such an approach, is deciding, from a sensor viewpoint, rather than a geographer's, which is field and which is object. From a purely GIS perspective the field/object solution is more semantic than image data content oriented; transcriptions of know objects in the world: mountains, rivers, roads…rather an *a priori* method of knowledge recording, provide a framework for ontology constructs: everyone knows a river, and there it is. However, the limitations of this world view were understood when, for example, one would try to decide where the very dynamic river object began and the river bank ended. This was difficult enough to ascertain during a ground survey, much less from overhead sensors looking at terrain during different times of year.

For modern intelligence gathering systems, finding and identifying a road can be accomplished, for the most part, automatically. However, finding a road that is more earth than road can be difficult, requiring perhaps special sensor configurations as well as special data processing. This is a case of the "object" merging with the "field". In fact, the entire problem of object recognition in sensed data can be reduced to first detecting the object,, that is, separating it from the background, and then recognizing what the object is and subsequently characterizing the target object [6]. Furthermore, tracking, or maintaining a view of the detection, a key capability for a video surveillance system, presupposes that 1) an object of interest has been detected and 2) the same object is being recognized in subsequent temporal increments: that is, being tracked.

While detection and tracking of objects in motion came into formal study during World War Two with the invention of RADAR, and the technical evolution of tracking since that time has of course been significant, yet the fundamental problems are the same. The issues have centered on state estimators, probability, statistics, and linear system analysis, all somewhat outside the scope of this paper. Yaakov Bar-Shalom portrays the problem as "…estimation of states of targets in surveillance systems operating in a multitatarget-multisensor environment. This problem is characterized by measurement origin uncertainty."[7]. However, once a system dominates uncertainty, target classification and population of ontological entity objects may proceed. Ontology refinement becomes a function of simply combining the extracted target objects with the collection associated metadata so that a vehicle image in one collect is differentiated from a vehicle image in a different collect. The fuzzy boundaries of the river-bank object can be quantified by a metadata structure with metrics appropriate to the target, or qualified by a time of year tag. Multiple target ambiguity is reduced in a sequence by noting position based on platform geoposition and camera pointing: information carried in the metadata stream [3].
In our preprocessing to populate ontology classes representing image content, we were able to successfully employ multiphase image decomposition and shape recognition algorithms [8] to extract target objects from video scenes. Local contextual information combined with statistical boosting was part of this image analysis process. Learning object representation is also an important part of the analysis and compatible with multiframe video so that subsequent collects of similar objects will enhance recognition success.

4 Integration of Content

The pivotal classes to be developed in the VideoIMINT ontology are the classes of imagery content and targets. Since both methods of extracting such imagery features: manual and automatic are utilized, an important aspect of the development effort was to define these classes and properties so as not preclude one or the other method while remaining consistent with other class and property descriptions.
In the ontology design, we have already constructed a class of *Property*, a *Dependent Continuant* of *Entity* class. The elements of *Property* include physical properties such as Location, Direction, Distance, Height, Length and other physical features. While these properties can apply to aircraft, sensors or other *Independent Continuant*, *Physical Entity* Classes, they can also apply to imagery content classes such as an Airfield *Control Tower*. Thus, new classes can be added to capture the concepts of imagery content – targets, and the existing property classes can be used to define them, dimensions and location.
The human analyst can efficiently recognize and tag content in videos, however, as was posited early in this paper, there is just too much to view. Therefore, automated extraction processing is an important mechanism for populating instances of the ontology while recognition of content detail and differentiation is not necessarily important. As was demonstrated in the development program, simple recognition of a "runway", a "control tower" facility and "aircraft" was sufficient to locate specific

video segments to vastly reduce the amount of data a human needed to review. The recognition of aircraft type was unnecessary, only the fact of presence of aircraft in the video made an enormous difference in video volume requiring human review. Figure 3 shows how segments of the video were highlighted by the ontology reasoner "knowing" that the class of *Aircraft* had been populated by the recognition engine. Those segments of identified video also reference the associated geospatial location, time, sensor and other details regarding the collection.

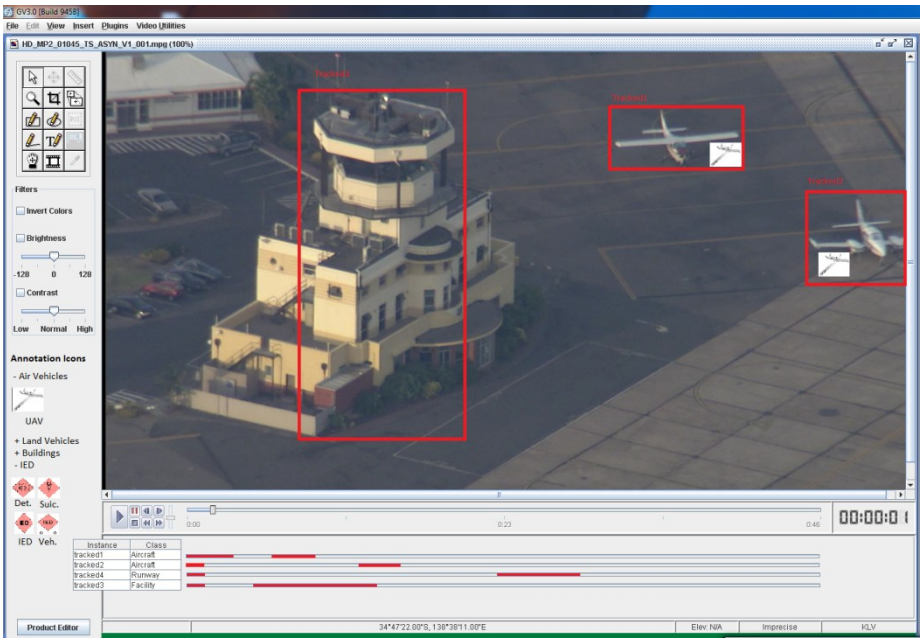


Figure 3. Video viewer showing highlighted segments of recognized content. The analyst has only to skip to that segment to find aircraft – and perhaps add his own tag of type identification.

5 Intelligence Assurance

A practical aspect of all this metadata information, along with the imagery (*InformationContentEntity*) is the inherent ability to determine quality of collection at any time, and conversely, the ability to predict collection performance *a priori*, in order to manage missions in terms of platform/sensor and operations to complete mission requirements and fulfill Essential Elements of Information (EEI) needs. That is to say, if the mission is to image an SA-6 Integrated Air Defense System (IADS) as opposed to determine whether individuals in an urban area are carrying Man Portable Air Defense systems (MANPADS), the proper combination of aircraft/sensor/altitude can be determined prior to mission execution: essentially a dynamic National Imagery Interpretability Rating Scale (NIIRS) for video collection to drive tasking.

Furthermore, imagery can be subject to valuation for quality metrics, such as consistent General Image Quality Equation (GIQE) [9] performance in regards to factors such as spatial resolution in terms of ground sample distance (GSD), relative edge response (RER) and overall system modulation transfer function (MTF) [10], after the fact, to determine system performance efficacy. All of these factors can be calculated, in many cases dynamically, but certainly as simple reasoned queries into the ontology. The true value is that the semantics of system performance are enforced by the ontology such that the variables of formula are consistent, yielding comparative and useful results. It is then possible to understand how one platform/sensor combination will perform, or is performing, relative to another under varying conditions for various missions.

6 Data Storage and Fusion

An integrated approach to video collection systems that includes processing, ontology mapping, storage and fusion would certainly enhance the overall utility and value of this intelligence source. Integration of an ontology with a tightly coupled storage system can yield value in the same manner as designing a data schema will for any data storage system. In fact, there are many similarities between a database schema and an ontology. However, one of the major differences is that a schema is essentially a static construct and does not support logical inferences in the way an ontology does. [11, 12] For example, a query into ontology might ask if a particular imaged runway can support a large cargo aircraft. The ontology can explore data rules regarding classes of runways, aircrafts and their properties, one of which may be a relationship between aircraft type with a property of landing *Distance (length)*, and *Weight* (*Load*) while the classes and subclasses of *Object*←*Facility*←*Airport*←***Runway*** will have a similar property of *Length* and another of *Load*. Thus, if a runway image falls into a particular runway ontology class, then the inferred condition that it will support certain aircraft is straight forward. The Database, on the other hand, has the explicit requirement of a schema entry to identify that runway has a certain characteristics as part of a data storage tuple, without inferring a particular aircraft can use that runway.

6.1 Storage Approaches

While in theory, the ontology for VideoIMINT could operate on any data video that was known to the ontology (*i.e.* standard video products); a tightly coupled storage system is more efficient. The ability to reference the storage system upon which the ontology operates is a great advantage. Short-term storage will make searches more efficient and rapid while longer term retrieval, the forensic search, can be enhanced as a class in the ontology with rules guiding which data is stored for what periods of time. Temporal redundancy, similar to information redundancy, can guide the "compression" of video for longer term, more efficient, storage if the storage rules operating on this data are clearly defined (semantically consistent). For example, a vehicle "track" can include the content of the tracked video as only a segment vector of the video frame through time. Utilizing a common method of video compression,

the “background” can be intermittent frames (I-frames, B-frames or P-frames of the MPEG specification) that maintain the slower changes in the surrounding scene. I t would be unnecessary to retain all the traditional I, B, or P-frames but rather only those frames useful to understand the context of the tracked target. Further compression is achieved by rendering these frames as wavelet compressed data according to the JPEG2000 compression schemes [13]. The track itself can be stored as a separate class of wavelet, type *Track*, with useful subclasses and properties. Regardless of scheme employed to store data from video, control of a short-term storage of data will enhance the operation of the ontology.

6.2 Data Fusion

When building an ontology of imagery content and associated metadata, these classes become the inputs for stipulated data fusion processing, at least for lower levels of the Joint Directors of Laboratories (JDL) Data Fusion Model (1998 revision) [14] that include Object Detection and Assessment and Object Refinement.
As targets are detected and assessed, declarations of object are made which in turn enables the population of ontological object classes (*e.g.* vehicle). The thresholds and rules governing this instancing are the same thresholds and rules that will (or will not) satisfy subsequent fusion processing of these detected objects. Associations of metadata, related to these instances will allow further Object Refinement in the sense of positioning, sizing and characterizing the ontological object thus enhancing fusion processes with associated metrics. Such qualifications will enable overall correctness of initial assessments in terms of accuracy, precision, and error within the fusion process.
Consider fusing two different collects of video data, from different sensor types, at different times covering a similar geographical location. The imagery must be collected, located, registered spatially and temporally, while the characteristics of the sensor, the look angle and altitude (for resolution purposes) all need to be considered to just begin the fusion process. However, the classes and properties that have been described previously in this paper do just that. Utilizing the metadata alone, almost all sensors and platforms provide this information, and it is rendered by the ontology into appropriate classes with properties. That information which is not collected, for example, pixel image resolution, can be readily calculated from sensor specification, sensor pointing data, and platform performance data, all readily available. The only other fusion requirement is that the ontology enforces semantic consistency of units and metrics. The fusion processes can now be built into the data processing chain with sensor selection tasking “switches” to choose appropriate sensors for a particular mission and appropriate systems operations. The data preconditioning for fusion is completed: leaving specific, mission related fusion processes, with inputs necessary for predictable, consistent sensor data fusion.
The construction of the ontology must however, consider such subsequent processing in the design of classes and properties. While the necessary metadata and class descriptions can be built, they may not be consistently populated from one sensor to the next of one collect to the next. We may provide the facility for the subsequent operation, which does not, however, guarantee fusion.

7 Summary

This prototype ontology construct for Video Imagery Intelligence collection demonstrated the value of integrating video metadata along with specification information and imagery content in an organized, semantically consistent structure based on standards. Additionally, direct logical queries into the ontology were able to identify video segments with tagged and extracted features and mark those segments for review by an analyst. The ontology structures appear to be a valuable and useful tool to bring under control the growing volumes of data that is being collected by Unmanned Aerial Vehicles in various mission circumstances. The ontologies, if developed correctly, can also be used as both a mission planning system and a dynamic control system based on proven approaches such as NIIRS guided tasking. Overall performance quality can be monitored in real time to ensure the efficient and effective operation of intelligence collection platforms.
Finally, the use of ontologies enforces a semantic consistency as well as maintenance of performance information that forms the basis of sensor data fusion. Using the information collected and categorized by the ontology promises to facilitate building new fusion processes based on simple class relationships such as location, dimensional information and sensor operational performance.

Acknowledgements

Mr. Ron Rudnicki of CUBRC and the National Center for Ontological Research for his guidance and help in developing the OWL ontology used in this project.
Mr. Todd Howlett of the Air Force Research Lab at Rome, NY for sponsoring this activity as part of AFRL research into MultiINT information systems understanding.

References

[1] A single, 16 Mpixel camera with twenty-four bytes per pixel (color) and a two hertz frame rate would generate about 455 Gbytes of data in ten minutes. Six such cameras on a platform would push well into the terabyte range in 10 minutes.

[2] http://www.nytimes.com/2010/01/11/business/11drone.html

[3] *Advanced Authoring Format Profile for Aerial Surveillance and Photogrammetry Applications*, Version 1.0, National Geospatial-Intelligence Agency Motion Imagery Standards Board, Washington D.C. January 8, 2006.

[4] *Ibid, UAV Datalink Local Metadata Set.*

[5] Câmara, G., Egenhofer, M., Fonseca, F., and Monteiro, A. M. V. *What's in an Image?* in: Montello, D. R., (Ed.), Spatial Information Theory—A Theoretical Basis for GIS, International Conference COSIT '01, Santa Barbara, CA.

[6] Waldman, G., Wootton, J., Electro Optics Systems Performance Modeling (pp190-192). Artech House, Norwood, MA. 1993.

[7] Bar-Shalom, Y., Li, Xiao-Rong, Multitarget-Multisensor Tracking: Principles and Techniques. Storrs, Connecticut, 1995.

[8] Corso, J.J., New York State University at Buffalo, Department of Computer Science and Engineering. 2010.

[9] Leachtenauer, J.C., Malila, W., Irvina, J., Colburn, L., Salvaggio, N. *General Image Quality Equation: GIQE*. Applied Optics, Vol.36, No. 32. November 1997.

[10] Granger, E.M., Cupery, K.N. *An optical merit function (SQF), which correlates with subjective image judgments*. Photographic Science and Engineering, Vol. 16, No. 3, May-June 1972.

[11] Horrocks, I., Ontologies and Databases, a W3C Presentation. Oxford University, 2010.

[12] Motik, B., University of Manchester; Ian Horrocks, Oxford University; Ulrike Sattler, University of Manchester, *Bridging the Gap Between OWL and Relational Databases*, World Wide Web Conference Committee, May 2007.

[13] *ISO/IEC 15444-1:2004 | ITU-T Rec. T.800* defines a set of lossless (bit-preserving) and lossy compression methods for coding bi-level, continuous-tone grey-scale, palletized colour, or continuous-tone colour digital still images.http://www.iso.org/iso/catalogue_detail.htm?csnumber=37674

[14] Hall, D., Llinas, J., Steinberg, A. N., Bowman, C.L., *Handbook of Multisensor Data Fusion*. pp **1**-7,8, **2**-5, ed. David Hall, James Llinas, CRC Press, LLC Boca Raton, Florida, 2001.

The Application of a Course-of-Action Ontology to Support OPFOR COA Selection and Assessment

Timothy Darr, Richard Mayer and Perakath Benjamin

Knowledge-Based Systems, Inc.
1408 University Drive East
College Station, TX 77840
{tdarr, rmayer, pbenjamin}@kbsi.com

Abstract. This paper describes the application of a course-of-action (COA) ontology to a demonstration scenario that the authors' company participated in that included a task to forecast the COAs to be executed by a simulated insurgent opposing force (OPFOR). The COA ontology includes standard decision-theoretic concepts to describe preference models from the perspective of an insurgent group for the purpose of predicting possible OPFOR COAs. The OPFOR preference structure is represented as a preference graph that visually displays the ranking of the COAs, from the perspective of the OPFOR decision maker, highlighting the most and least preferred COAs.

Keywords: Course of action planning, decision theory, utility theory, ontology, preference modeling.

1 Introduction

This paper describes the application of a course-of-action (COA) ontology [1,2] to a demonstration scenario that the authors' company participated in that included a task to forecast the COAs to be executed by a simulated insurgent opposing force (OPFOR). The COA ontology applies to the COA planning processes defined for the United States Army and Marine Corps for multiple domains, to include stability operations planning [3], counterinsurgency planning [4] and information operations planning [5]. The core ontology includes definitions of the common concepts and properties for defining COA plans, including: COAs, COA activities, COA phases, measures-of-performance (MOP) and measures-of-effectiveness (MOE).

To illustrate the use of the COA ontology, we use a scenario that is inspired by the Empire Challenge ’10 (EC 10) demonstration held at Fort Huachuca in August 2010 [6-8]. In this scenario, a coalition force (CF) unit is engaged in stability operations in an area that is under contention by an insurgent OPFOR whose goals include the following:

- Short-Term: inflict damage on the CF
- Medium-Term: discourage confidence in the host nation government

- Long-Term: establish religious and cultural control

In the area of operations (AOR) under contention, there are two tribal groups whose interests conflict with each other. Tribe A is generally supportive of the CF and tribe B is generally supportive of the insurgency. These loyalties are motivated in part by a long-standing set of grievances between the tribes: tribe A and tribe B do not like each other very much, but have negotiated an uneasy truce at the moment. There are no hostilities at this time, but there is a risk that hostilities could re-emerge at any time.

Fig. 1 shows the larger operational context in which the COA ontology is used. The Hidden Enemy Network Influence Operations Map (HENIOMAP) is an application under development that is used by decision makers to assess and forecast possible COAs to aid in their own planning. The output COA ranking is an ordering of the possible COAs given as input that clearly shows the most and least preferred COAs that are consistent with a preference model. The COAs that are provided as input can be either own force or OPFOR COAs. The preference model represents the trade-offs over multiple, conflicting attributes that a decision maker employs to select the most-preferred COA to achieve their goals. The HENIOMAP ontology, of which the COA ontology described in this paper is a component, defines the domain under consideration. The HENIOMAP algorithms are used to generate the COA rankings.

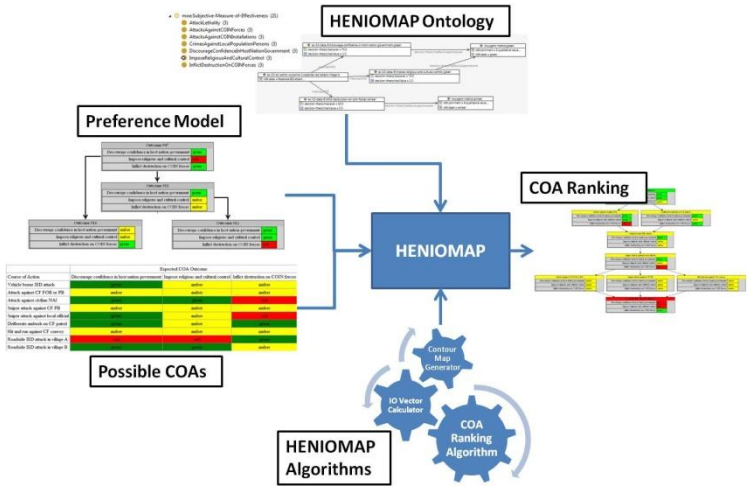


Fig. 1. Operational context for the COA ontology

Of the elements shown in **Fig. 1**, this paper describes part of the HENIOMAP ontology (COA ontology), the possible COAs, the preference model and a visualization of the COA ranking as used in the EC 10 demonstration. The details of the HENIOMAP algorithms are out of scope for this paper. Section 2 provides an overview of utility theory and a representation of some of the concepts in utility theory that are used to model COA selection and assessment problems. Section 3

provides an overview of the COA ontology that was used in the EC 10 demonstration. Section 4 presents some conclusions.

2 Utility Theory Overview

Before describing the COA ontology and how it supports the example scenario, we provide an overview of utility theory. The COA ontology models OPFOR COA selection and assessment as a multi-objective decision problem. Given a broad overall objective to be realized by a specific operation, the decision-maker must select the "best" COA to perform to achieve some objective or to identify the "best" outcome to try to achieve via a sequence of actions, where "best" is defined as an outcome that satisfactorily trades the conflicting objectives against one another from the perspective of a given decision maker.

Utility theory was originally developed in economics to measure the desirability of a good or alternative from the perspective of an agent [9]. In this model, a COA outcome replaces a "good or alternative" in the economics application, and a COA planner or decision maker replaces an "agent" in the economics application.

A utility function is given by

- $u: O \rightarrow \mathbb{R}$, where $O$is a COA outcome and $\mathbb{R}$is a real-valued number

A common form of utility function is a weighted sum of attribute values

- $u(o_i) = \sum_k w_k * a_k[o_i]$, where $w_k$is an attribute weight, and $a_k[o_i]$ is an attribute-value score that assigns a real-valued number to the attribute value $a_k$for outcome o_i. The weights of each attribute are assigned by an SME / analyst or by using an algorithm to estimate the weights
  - In this domain, the $a_k[o_i]$is a numeric value assigned to a goal that describes a COA outcome

A preference is a relation between two outcomes such that $u(o_i) \geq u(o_j)$.

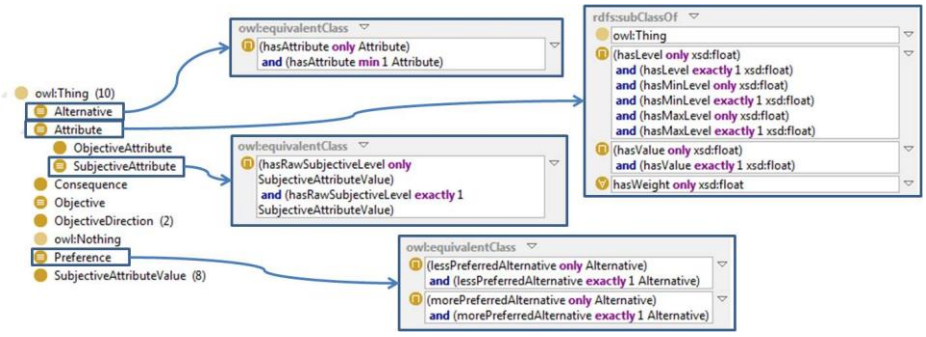


Fig. 2. Decision theory concepts used in the COA ontology

Fig. 2 shows concepts in the COA ontology that represent the elements of the utility-theoretic definition. The alternative class corresponds to the utility theory outcome and is described by a collection of attributes. The attribute class corresponds

to the utility theory attributes and are described by attribute levels, utility-theoretic values and weights. The preference class corresponds to the utility theory preference and is a relation between alternatives in which one alternative is preferred to another attribute, from the perspective of a given decision maker. In addition, a subjective attribute class is added as a subclass of attribute to represent attributes whose values are non-numeric. Extensions of these concepts for the COA ontology, along with examples, will be given in the next section in the context of the EC 10 demonstration.

3 Course-of-Action Ontology to Support EC 10

This section describes extensions to a COA ontology for counterinsurgency operations [1, 2] to support the EC 10 demonstration.

Fig. 3 illustrates a COA outcome forecast table, which shows in stoplight format the possible OPFOR COAs. In this scenario, the CF commander uses this table to help make decisions on his own COAs based on what the OPFOR is likely to do.

	Expected COA Outcome		
Course of Action	Discourage confidence in host nation government	Impose religious and cultural control	Inflict destruction on COIN forces
Vehicle borne IED attack	green	amber	amber
Attack against CF FOB or PB	amber	amber	amber
Attack against civilian NAI	green	green	red
Sniper attack against CF PB	amber	amber	amber
Sniper attack against local official	green	amber	red
Deliberate ambush on CF patrol	green	amber	green
Hit and run against CF convoy	amber	amber	amber
Roadside IED attack in village A	red	red	green
Roadside IED attack in village B	green	green	amber

Fig. 3. The COA effect forecast table shows the expected or forecast change of state for a given COA using a red / amber / green indication

The rows in this table represent the possible OPFOR COAs, and the columns represent the assumed goals of the OPFOR. The effect of a COA for each goal is shown as red / amber / green indicators, where green indicates the best possible outcome for the OPFOR, red is the worst possible outcome the OPFOR, and amber lies somewhere in between. The indicators for each of the goals are *from the perspective of the OPFOR*. The COA effect table is created from historical data mining, manual entry by an SME or analyst, or more likely some combination of automated mining and manual entry[1].

In this particular scenario, we assume that the current state is amber for the medium- and long-term goals "discourage confidence in the host-nation government" and "establish religious and cultural control" and red for the short-term goal "inflict damage on the CF". In this state, there is room for improvement for the long-term

[1] Note that these predictions can be highly subjective and it is possible that different SMEs or analysts will come up with different forecast effects

goal and room for improvement or degradation for the medium- and short-term goals, from the perspective of the OPFOR. For example,

- If the OPFOR chooses to execute a vehicle-borne IED attack (VBIED), then it will improve its goals to inflict damage on the CF (red to amber) and discourage confidence in the host-nation government (amber to green), but will neither improve nor degrade its goal to establish religious and cultural control.
- If the OPFOR chooses to execute an attack against a CF forward operating base (FOB) or patrol base (PB), then it will improve its goal to inflict damage on the CF (red to amber) and not impact any of the other OPFOR goals.

Fig. 4 illustrates the COA preference graph, which is a representation of the preference structure of a decision maker. Each node in the graph represents a possible COA and edges represent the "is-preferred-to" relation. An edge from a source node to a target node indicates that the source node is preferred to the target node, from the perspective of a decision maker. The contents of the node show the forecast effects of the corresponding COA, as described above.

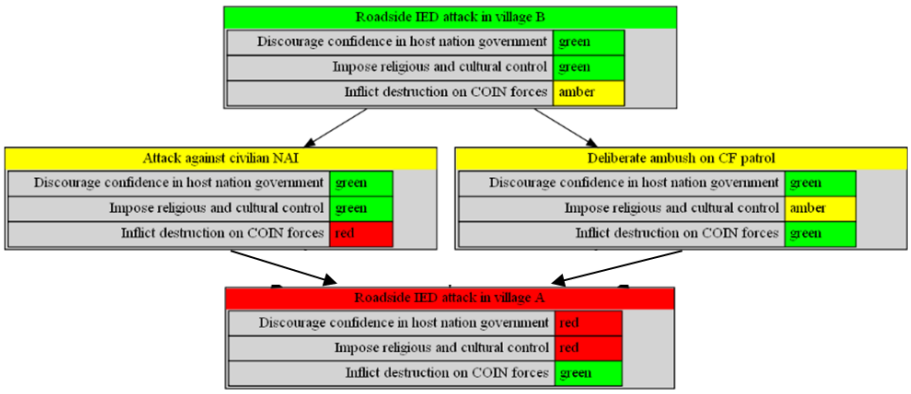


Fig. 4. The Preference Graph shows the preference structure for a collection of COAs from the perspective of a specific decision maker.

The colored title bar of each node represents the overall preference of each COA: a green title indicates that the COA is the most preferred; a red title indicates that the COA is the least preferred; and an amber title indicates that the COA is neither the most nor the least preferred. By inspection, a commander is able to visualize the most- and least-preferred COAs. Using this graph, the own force commander can assess which COA the OPFOR is likely to pursue and take that into account when formulating the blue force own COAs.

Fig. 5 shows the attributes and COA outcomes to support the EC 10 demonstration. The objectives of the OPFOR, shown in the upper left of the figure, are modeled as utility-theoretic attributes by extending the ontology described in section 2. These attributes include the short-term, medium-term and long-term goals described in section 1. The outcome of an OPFOR action, shown in the upper right of

the figure, is modeled as a decision-theory alternative. The key ontological modeling decision represented here is that the outcomes of the OPFOR actions are the outcomes over which the decisions are made. For EC 10, the decision is an assessment of which outcome is the most preferred COA for the OPFOR, given the objectives of the OPFOR decision maker.

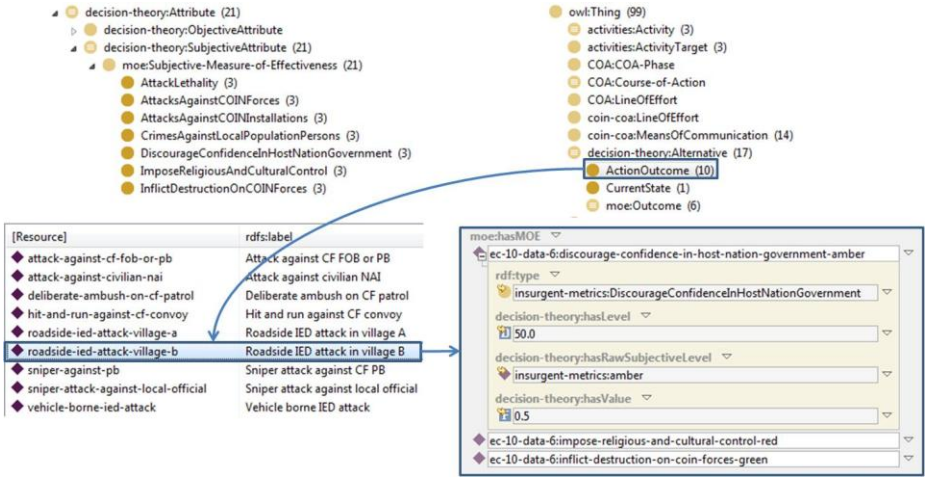


Fig. 5. Course of action ontology concepts to support the EC 10 demonstration

Individual outcomes are shown in the lower left of the figure as instances of the COA action outcome class. An example of the outcome that results from a vehicle borne IED attack (VBIED) is shown in the lower right of this figure. The outcome is described in terms of the attributes "discourage confidence in host nation government" (the medium term goal), "impose religious and cultural control" (the long-term goal) and "inflict destruction on CF forces" (the short-term goal).

The description of a COA outcome attribute is described by an attribute level, a raw subjective level, and a value. The attribute level is the source measurement of the attribute; for example, the "discourage confidence in host nation government" attribute might be measured by the rate at which the local population goes to the host-nation government to resolve legal disputes or obtain loans or other financial assistance, instead of going to the shadow insurgent government. The lower the rate, the better for the insurgents.

To support decision making in the context of the EC 10 demonstration, two rules are necessary to convert the attribute levels to a raw subjective level and to convert the raw subjective level to a utility-theoretic value.

Fig. 6 shows the SPIN[2] rule for assigning a raw subjective level (green / amber / red) to a given raw level band. The first and second arguments to the rule define the raw level band bounds and the third argument is the raw subjective level for those bounds. For this rule for the EC 10 demo, each of the attributes were assigned a raw

[2] http://spinrdf.org

subjective level of red for attribute levels less than 33% (in the rule, arg3 = red, arg1 = 0.0, arg2 = 33.0); green for attribute levels above 66% (in the rule, arg3 = green, arg1 = 66.0, arg2 = 100.0); and amber for all other attribute levels (in the rule, arg3 = amber, arg1 = 33.0, arg2 = 66.0). For the example shown in **Fig. 5**, the attribute level for the attribute "discourage confidence in host-nation government" for the action outcome "roadside IED attack in village A", the attribute is 50, so the rule would assign a raw subjective level of amber.

```
CONSTRUCT {
   ?this decision-theory:hasRawSubjectiveLevel ?arg3 .
}
WHERE {
   ?this decision-theory:hasLevel ?hasLevel .
   OPTIONAL {
      ?this decision-theory:hasRawSubjectiveLevel ?existingRawLevel .
   } .
   FILTER (!bound(?existingRawLevel)) .
   FILTER ((?hasLevel <= ?arg2) && (?hasLevel >= ?arg1)) .
}
```

Fig. 6. SPIN rule for assigning a raw subjective level (green, amber, red) for a given attribute level band

```
CONSTRUCT {
   ?this decision-theory:hasValue ?hasLevel .
}
WHERE {
   ?this decision-theory:hasRawSubjectiveLevel ?level .
   FILTER (?level = ?hasRawSubjectiveLevel) .
}
```

Fig. 7. SPIN rule for assigning a value ([0.0 1.0]) for a given raw subjective level

Fig. 7 shows the SPIN rule for assigning a utility value to a raw subjective level (green / amber / red). The first argument to the rule (hasLevel) is the utility value for the second argument, the raw subjective level (hasRawSubjectiveLevel). For this rule for the EC 10 demo, a raw subjective level of red was assigned a utility value 0.0 (in the rule, hasLevel = 0.0, hasRawSubjectiveLevel = red); green was assigned a utility value 1.0 (in the rule, hasLevel = 1.0, hasRawSubjectiveLevel = green); and amber was assigned a utility value 0.5 (in the rule, hasLevel = 0.5, hasRawSubjectiveLevel = amber). For the example shown in **Fig. 5**, the raw attribute level for the attribute "discourage confidence in host-nation government" for the action outcome "roadside IED attack in village A" is amber, so the rule would assign a utility value of 0.5.

4 Conclusions

Initial results from the Empire Challenge 10 demo showed promise for the approach described in this paper, especially the COA effect table and preference graph. Potential users were able to clearly assess the changes to state for each OPFOR COA as well as the most- and least-preferred COAs for the OPFOR decision maker. The COA ontology, augmented with concepts from utility theory, provides a strong theoretical foundation for creating the preference graph and using it as a COA assessment tool.

While utility theory has been use in modeling decision problems similar to COA assessment and selection, the marriage of the utility-theoretic model and semantic technologies has conferred the following benefits:

- The underlying utility theory model provides a theoretically sound foundation for defining useful properties and rules to support COA selection and assessment; these properties and rules are easily modeled using semantic technologies
- The ability to transform raw data, using a handful of simple SPARQL rules, into RDF-based representations to support visualizations that are natural for military decision makers; for example, the red / amber / green visualizations in the COA effect table.
- The ability to quickly modify the preference structure of a decision maker in a dynamic environments.

Acknowledgements: This work was supported by the Office of Naval Research under Contract N00014-09-C-0334.

5 References

1. Darr, T. P., Benjamin, P. and Mayer, R., "Course of Action Planning Ontology", Ontology for the Intelligence Community Conference (OIC 2009), George Mason University, October 21-23, 2009.
2. Darr, T. P., Benjamin, P. and Mayer, R., "Course of Action Ontology for Counterinsurgency Operations", 15th International Command and Control Research and Technology Symposium (CCRTS), Santa Monica, CA, June 22-24, 2010
3. "Stability Operations", Headquarters of the Department of the Army, Field Manual No. 3-07 (FM 3-07), October 2008.
4. "Counterinsurgency", Headquarters Department of the Army, Field Manual No. 3-24 (FM 3-07), Headquarters Marine Corps Combat Development Command Department of the Navy, Marine Corps Warfighting Publication No. 3-33.5 (MCWP 3-33.5), December 2006.
5. "Information Operations: Doctrine, Tactics, Techniques and Procedures", Headquarters Department of the Army, Field Manual No. 3-13 (FM 3-13), November 2003
6. "Empire Challenge 10", Date Accessed: August 27, 2010, Available: http://www.jfcom.mil/about/fact_ec10.html.
7. "EC 10: Green Devil provides improved tracking", Date Accessed: August 27, 2010, Available: http://usjfcom.dodlive.mil/2010/08/10/ec-10-green-devil-provides-improved-tracking/.
8. "Empire Challenge 10 finishes up", Date Accessed: August 27, 2010, Available: http://www.jfcom.mil/newslink/storyarchive/2010/pa081410.html.
9. Keeney, R.L. and Raiffa, H. "Decisions with Multiple Objectives: Preferences and Value Tradeoffs", Wiley and Sons, New York, 1976

A Semantic Wiki Alerting Environment Incorporating Credibility and Reliability Evaluation

Brian Ulicny[a], Christopher J. Matheus[a], Mieczyslaw M. Kokar[a,b]
[a]*VIStology, Inc.* [b]*Northeastern University*

Abstract. In this paper, we describe a system that semantically annotates streams of reports about transnational criminal gangs in order to automatically produce models of the gangs' membership and activities in the form of a semantic wiki. A gang ontology and semantic inferencing are used to annotate the reports and supplement entity and relationship annotations based on the local document context. Reports in the datastream are annotated for reliability and credibility in the proof-of-concept system.

Keywords: media monitoring; semantic analysis; entity/relation extraction; event tracking; gangs; reliability; credibility

1 Introduction

In this paper, we describe a prototype we are developing that we call the Semantic Wiki Alerting Environment (SWAE). SWAE ingests streams of open-source news media and social media and automatically constructs a model of transnational criminal street gangs, including their membership and their activities. The system automatically provides updates and alerts to significant changes in that model in the form of emails, text alerts and semantic wiki pages. The system relies heavily on ontology-based semantic annotation [1].

In today's intelligence and battlespace environment, large amounts of data from many sources must be effectively analyzed in a timely manner in order to provide an accurate and up-to-date understanding of current and potential threats. Key to understanding these threats is the identification and characterization of the various entities that they involve. These include the relevant individuals, groups, locations and events along with their corresponding interrelationships.

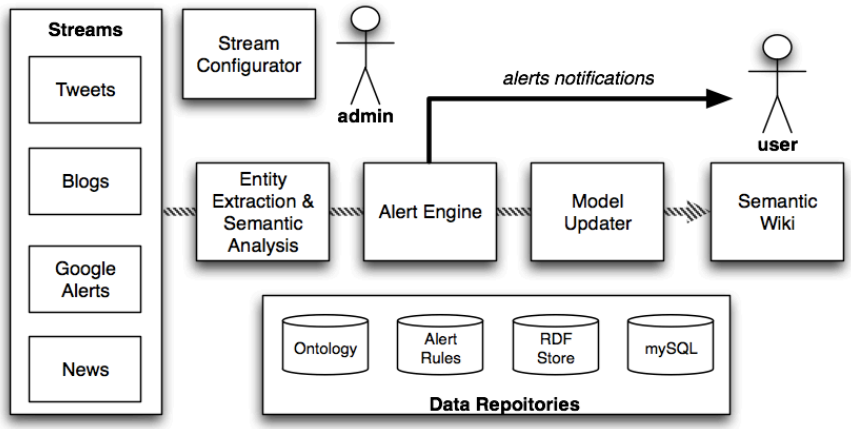
A wiki is a Web-based environment in which users can easily edit the text and layout of documents using a simplified, non-HTML syntax. Wikipedia is the most familiar example: a world-wide encyclopedia that any user can edit. Change-tracking by author and automatic hyperlinking are important aspects of wiki functionality. A semantic wiki is a wiki in which users can not only easily insert hyperlinks between documents, but in which semantic annotations of documents can be easily edited by users. In a semantic wiki, semantic web triples are encoded directly in the text. The subject of the triple is the topic of the page itself; predicate and object are then encoded as attribute::value pairs in the text. Thus, the markup [[population::3,396,990]] on a page for *Berlin*, asserts that Berlin has a population of that size. One can further represent that the *population* predicate is of Type::Number, to enable proper sorting and comparison. These triples can be used within semantic queries and to populate visualizations such as maps, timelines, and graphs automatically. We use the Semantic MediaWiki platform, an extension of the MediaWiki platform that underlies Wikipedia.

In their current state, semantic wikis are relatively primitive and require significant human effort in order to annotate the wiki's contents with semantic markup consistently [5]. However, in this project we have customized a semantic wiki to automatically pre-process incoming data from multiple sources, extracting relevant semantic information (explicit metadata and implicit relationships) and rendering it in a form readily consumable, and editable, by human analysts through the wiki interface. We also implement user definable alerting capabilities to permit automated notification regarding significant new events or critical changes in the composite representations of key entities such as dangerous individuals or groups. Ontology-based alerting capabilities of this sort necessitate the use of a formal inference engine, ideally one that is rule based to facilitate and simplify user customization.

2 System Overview

The high level design of SWAE is depicted in Figure 1. Data flows into the system from the left in the form of data streams (e.g. Tweets (Twitter updates), Blogs, news, alerts (standing news queries)). These reports are processed by the entity and relation extraction and semantic analysis algorithms. The annotated results are placed into the data repository and trigger the invocation of the alert engine, which is based on the SPARQL query engine available in the Open Sesame RDF data store. The results are used to inform the user of significant items and to update the semantic model maintained in the semantic wiki for subsequent access and further analysis by users. Semantic wiki pages are created automatically from the RDF produced during semantic analysis and entity extraction.

Figure 1. SWAE Data Flow



For development purposes, we have chosen to monitor data about the activities of transnational gangs as the focus of our investigation. There are many parallels between countering organized gang activity and counterinsurgency. Reports about gang activities are readily available from open sources and do not require translation.

We monitor several RSS feeds and periodically download and process new items in order to update the system. In addition we track news media outlets and law

enforcement press releases that we obtain via the news aggregator service Topix.net. Social media platforms such as Twitter (twitter.com), and Flickr (photo sharing) contain many reports by both self-professed gang associates and those chronicling their activity; these data streams, however, are quite noisy. Twitter status updates mentioning gang names contain a mix of chatter about the gang, unrelated uses of the term and links to news articles. Photo sharing sites such as Flickr (flickr.com) contain many depictions of gang graffiti, which can often be mapped to specific times and locations; several groups on Flickr are dedicated to documenting gang graffiti.

Our goal is to monitor these social media and open-source media streams in order to trigger alerts such as:

- A 10% increase in gang G's weekly incidents of type I in location L
- First occurrence of incident I by G in L in past year
- A 10% increase in attacks of G1 on G2
- New member of gang G
- A 10% increase in G membership in L since T
- New leader L of G
- A 20% increase in communications between members of G in past 24 hours
- Social media report of gang activity not correlated with media report
- Graffiti by or about gang G .

3 Ontology

In the Street Gang ontology (Figure 2) there are four primary top-level classes: Organization, Person, Incident and Information. The ontology defines numerous types of Incidents but distinguishes between CriminalIncidents and non-criminal incidents, the former of which is used to infer members of the Criminal class. There are also several types of Information corresponding to the source data that SWAE processes. There are two secondary classes - IncidentRate and Source. IncidentRate is intended to be used to record information about the *count* of incidents of a certain *incidentType* that are carried out by an Organization in a given period of time; these elements were added as it became clear from our sample rules that such constructs but be necessary to support many of them. There is also a Source class that was created to permit the author of a piece of Information to be either an Organization or a Person.

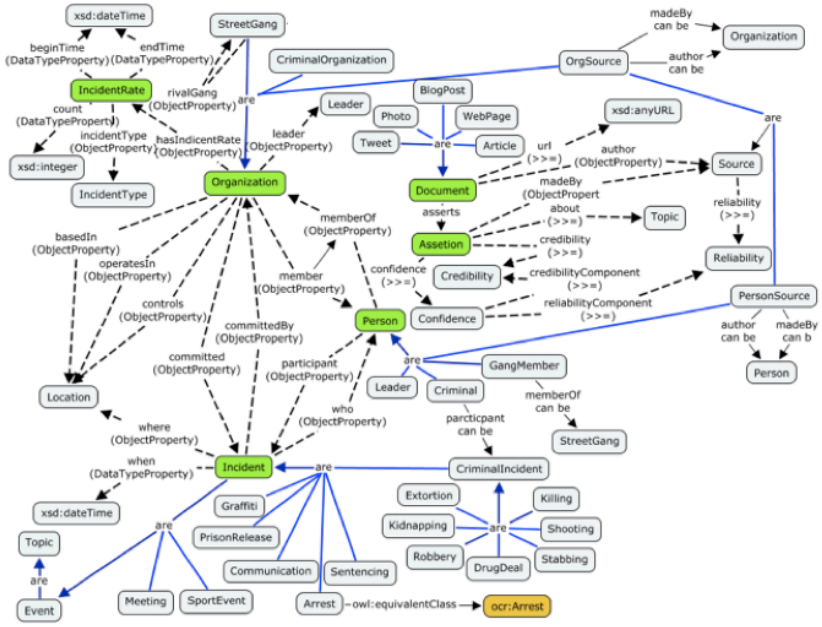
4 System Components

Feeds for data sources are periodically re-queried in order to obtain the latest reports from both media outlets (which are analogous to analyzed intelligence reports) and social media reports such as Tweets and Flickr photos (which, if not citing media outlets, are analogous to source material that has not yet been subject to intelligence analysis). Feeds in non-RDF compliant formats are converted to RDF automatically. These source feeds provide useful metadata about the reports. Links from the RSS feeds are automatically extracted and are then processed using the OpenCalais API[1] to extract basic level objects and relations based on their local context in the text.

In the following extract, OpenCalais's output detects the presence of an arrest relationship in the string

[1] OpenCalais Web Service API. http://www.opencalais.com/calaisAPI

Figure 2 Gang Ontology



…The arrest follows the May 28 arrest in Santa Cruz of [X], another [Gang Y] member…

The RDF output of Open Calais encodes the detection of an instance of the Arrest relation as follows: an entity of type InstanceInfo is created with URI "…/Instance/40". This InstanceInfo is about (oc:subject) URI. Note that this InstanceInfo doesn't by itself provide explicit information about who was arrested, when or where. (The 'oc' prefix denotes an OpenCalais namespace.)

```
  <rdf:Description rdf:about="http://d.opencalais.com/dochash-
1/6d3695ba-2142-3679-b8db-5e206844c924/Instance/40">
    <oc:detection>   <![CDATA[[in a news release.</p><p> The arrest
follows ]the May 28 arrest in Santa Cruz of [X] [, another [Gang Y], or
[VariantName V], member]]]></b:detection>
    <oc:docId rdf:resource="http://d.opencalais.com/dochash-1/6d3695ba-
2142-3679-b8db-5e206844c924"/>
    <oc:exact>the May 28 arrest in Santa Cruz of [X]</b:exact>
    <oc:length>55</b:length>
    <oc:offset>1071</b:offset>
     <!—this incident URI is what the InstanceInfo is about->
    <oc:subject rdf:resource="http://d.opencalais.com/genericHasher-
1/f6e310ea-f54a-3aee-bc99-7293eea20f44"/>
     <rdf:type
rdf:resource="http://s.opencalais.com/1/type/sys/InstanceInfo"/>
  </rdf:Description>
```

This further part of the OpenCalais output says that the indicated incident is of rdf:type oc:Arrest. It also specifies that the oc:person of the Arrest incident is specified by the URI indicated. It also indicates the date string ("May 28") and normalized date (2010-05-28) of the incident. Note that this RDF snippet about the incident URI (i.e. all the information about this incident in RDF form) doesn't specify, in particular, where the Arrest took place. None of the elements below that are associated with the Arrest incident are guaranteed to be present in the OpenCalais output depicting an incident of rdf:type Arrest.

```
  <!– The same Incident URI identified in the InstanceInfo RDF->
  <rdf:Description rdf:about="http://d.opencalais.com/genericHasher-
1/f6e310ea-f54a-3aee-bc99-7293eea20f44">
       <rdf:type
rdf:resource="http://s.opencalais.com/1/type/em/r/Arrest"/>
       <c:person rdf:resource="http://d.opencalais.com/pershash-
1/1b1289ef-845f-31a9-a640-b6724dbe61e1"/>
       <c:date>2010-05-28</c:date>
       <c:datestring>May 28</c:datestring> </rdf:Description>
```

5 Semantic Analysis

OpenCalais' processing is very sophisticated, but because it does not always specify what we need it to specify for our alert processing, we need to do semantic analysis of the RDF graph and the original text in order to both augment and correct the RDF that has been output. OpenCalais' recognizes entities and relationships based on their local context only; we often need to use global- or document-level inferencing to determine other relationships and entities.

We use the VIStology-developed inference engine, BaseVISor[2], to modify and augment the RDF produced by OpenCalais, and save the modified RDF. BaseVISor is VIStology's forward-chaining inference and rule engine that infers facts from an RDF/OWL store based on an ontology (using OWL 2 RL) as well as user-specified rules that can involve procedural attachments for things like computing the distance between two latitude/longitude pairs. BaseVISor has been optimized to process triples very efficiently.

This semantic processing by BaseVISor results in a number of augmentations to the data. First, the OpenCalais RDF output lacks datatypes on elements, so these must be supplied for integers, dates and other datatypes used in OpenCalais output. Second, we use BaseVISor rules to correct systematic misidentifications that OpenCalais makes. For example, OpenCalais always identifies one particular gang name as a Person, not as a variant name for a specific gang. These revision rules are necessary because end users cannot customize OpenCalais with a custom vocabulary at present. Third, we employ BaseVISor rules to make rule-based inferences about the text in order to supplement OpenCalais's event representations. As noted above, while OpenCalais identifies Arrest-type incidents in texts, it does not always identify the *who, what, where*, and *when* attributes of these events presumably because they can't be determined by the local context. We use BaseVISor rules to infer times and locations for the surrounding event based on the entire text. For example, if no

[2] VIStology BaseVISor Inference Engine. http://vistology.com/basevisor/basevisor.html

location is specified for an event in the OpenCalais RDF output, to a first approximation, we specify the closest instance of a City in the text as the location of the Arrest. Similarly, if no date for an arrest is specified, then we take the date of the report itself as the arrest date, and so on.

We also use BaseVISor to insert RDF triples for instances of types of things not identified by OpenCalais, such as the names of gangs, and to associate persons with gangs based on the OpenCalais RDF. For example, if OpenCalais specifies that there is a *joining* relationship, and the subject of the joining is a certain person, and the object of the joining event is "the ABC prison gang", then based on the presence of the term "ABC" in the object, we assert an association between the person and the ABC gang.

BaseVISor is also used to infer relations that are implicit from the data and the ontology as explicit triples. For instance, if the ontology says that "ABC" is a Gang, then if John is a member of the ABC gang, he is a gang member. A triple encoding this fact will be inferred and imported into the RDF store. All of the triples that can be inferred by means of these semantic analysis rules and the combination of the RDF output and the OWL ontology, using OWL 2 RL, are inserted into the global fact base.

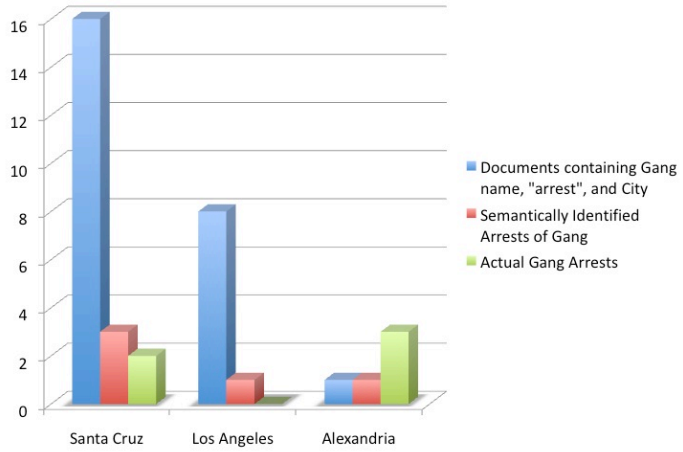
Finally, based on the OpenCalais RDF graph, we make API calls to other data sources in order to augment the RDF data store with the necessary data for querying. Although OpenCalais sometimes provides resolved geolocations for spatial entities like cities, it does not always do so. For instance, OpenCalais may identify "Santa Cruz" as being an instance of rdf:type City, but it does not always specify that this mention of a City actually refers to "Santa Cruz, California" with the corresponding latitude and longitude. Because OpenCalais cannot be forced to make a guess for every instance of City, we invoke the Geonames.org API in order to determine the latitude and longitude of the city based on document source metadata, from the feed.

After this, the data gathered and processed by the extraction component is imported into an OpenSesame RDF store and queried via SPARQL in order to update the model of the gang organization: its members, incident rates, event times and locations, and so on. The RDF data that has been input into the data store is periodically queried to provide semantic alerts, which are sent as email messages or text messages. Additionally, SPARQL queries are used to create and update topical pages in the Semantic MediaWiki reflecting our current knowledge of a gang.

6 Example and Discussion

In Figure 3, we contrast the approach outlined with more traditional keyword-based approaches to alerting and event tracking. The blue column shows the number of news stories containing a specified gang name, the name of the indicated city, and "arrest" over a two-week period in June, 2010. There were 16 documents corresponding to Santa Cruz, eight to Los Angeles, and one to Alexandria. Based on document counts alone, then, one would suppose that there were far more arrests in California than in Virginia during that period. However, the red column shows that automated semantic analysis identifies three arrests in Santa Cruz and one each in Los Angeles and Alexandria, VA. These are much closer to the actual figures (two in Santa Cruz; zero in Los Angeles; three in Alexandria, VA).

Figure 3 Event Counts: Keywords vs Semantic Analysis



The result of the semantic processing shows promise in that the total number of arrests identified per city is much closer to the actual result than one would infer from the document counts. Three arrestees out of four are correctly identified out of eighteen news articles (precision = 75%), and four out of six total arrestees in the corpus are identified (recall = 66%).

7 Information Evaluation

NATO STANAG (Standard Agreement) 2022 "Intelligence Reports" states that where possible, "an evaluation of each separate item of information included in an intelligence report, and not merely the report as a whole" should be made. It presents an alpha-numeric rating of "confidence" in a piece of information (compare [9]) which combines an assessment of the reliability of the source of the information and an assessment of the credibility of a piece of information "when examined in the light of existing knowledge".[3] The alphabetic Reliability scale ranges from A (Completely Reliable) to E (Unreliable) and F (Reliability Cannot Be Judged). A similar numeric information credibility scale ranges from 1 (Confirmed by Other Sources) to 5 (Improbable) and 6 (Credibility Cannot Be Judged).

As a first approximation, we have implemented some crude, initial rules for reliability and credibility. For example, if a source is from Topix (a news source) we mark it B (Usually Reliable). We could potentially mark reports from official government sources, such as FBI press releases, even higher. If a source is from Twitter or Flickr, we mark it 6 (Reliability Cannot Be Confirmed).

[3] North Atlantic Treaty Organization (NATO) STANAG (Standardization Agreement) 2022 (Edition 8) Annex. Essentially the same matrix is presented as doctrine in Appendix B "Source and Information Reliability Matrix" of US Army FM-2-22.3 "Human Intelligence Collector Operations" (2006), although STANAG 2022 is not explicitly cited.

For credibility, if two reports identify the arrest/trial/conviction/killing of the same person, we mark each such report as 1 (Confirmed By Other Sources). STANAG 2022 does not prioritize coherence with the earliest reports; rather, it says that the largest set of internally consistent reports on a subject is more likely to be true, unless there is contrary evidence. It is a military truism that "the first report is always wrong" [6], so a bias towards coherence with the first report on a subject should be avoided. Further research is need to determine the degree to which two reports must be similar in order to count as independent confirmation of one another.

8 Conclusion and Future Work

We have described a proof-of-concept system for automatically creating and updating a model of a group (here, a criminal gang) in the form of a semantic wiki. We incorporate into this model a preliminary implementation of the STANAG 2022 metrics for source reliability and information credibility. Initial work on this system presented interesting design decisions that we outlined here, along with plans for future work. Our work differs from other available systems in that it attempts to create and maintain a usable model of a group and its activities automatically by creating semantic wiki pages that represent the current state of knowledge of the group. Significant changes in this model are sent as email or text alerts to concerned parties. By normalizing references to entities, relations and events across documents, the system provides a solution to the problem of data redundancy in reports. In ongoing work, we plan to investigate the incorporation of social-network metrics of centrality as proxies for estimating source reliability [7], and to incorporate social-network measures of source independence into our credibility calculation.

Acknowledgments. This material is based upon work supported by the United States Navy under SBIR Award No. N00014-10-M-0088.

References

1. W3C OWL Working Group. http://www.w3.org/2007/OWL/wiki/OWL_Working_Group
2. Model Driven Architecture and Ontology Development, by Dragan Gasevic, Dragan Djuric and Vladan Devedzic, Springer publisher, 2006.
3. Wikipedia entry for Wiki. http://en.wikipedia.org/wiki/Wiki
4. Semantic Web Technologies – Tends and Research in Ontology-based Systems, by John Davies, R. Studer and P. Warren, Wiley publisher, October 2007.
5. J. Bao. The Unbearable Lightness of Wiking – A Study of SMW Usability. Presentation. Spring 2010 SMW (Semantic MediaWiki Conference). MIT. http://www.slideshare.net/baojie_iowa/2010-0522-smwcon
6. LTG (Ret) Ricardo S. Sanchez, Military Reporters and Editors Luncheon Address. 12 Oct 2007. http://www.militaryreporters.org/sanchez_101207.html
7. Ulicny, B., Matheus, C., Kokar, M., *Metrics for Monitoring a Social-Political Blogsophere: A Malaysian Case Study*. IEEE Internet Computing, Special Issue on Social Computing in the Blogosphere. March/April 2010.
8. Semantic MediaWiki. http://semantic-mediawiki.org
9. Schum, D., Tecuci, G., Boicu, M., Marcu, D., Substance-Blind Classification of Evidence for Intelligence Analysis, in Proceedings of "Ontology for the Intelligence Community," George Mason University, Fairfax, Virginia. October 2009.

Submitted to *The Fifth International Conference on Semantic Technologies for Intelligence, Defense, and Security* – STIDS 2010.

TIACRITIS System and Textbook: Learning Intelligence Analysis through Practice

Gheorghe Tecuci[1], Mihai Boicu[1], Dorin Marcu[1], David Schum[1]

Benjamin Hamilton[2]

[1] Learning Agents Center, George Mason University, Fairfax, VA, USA
{ tecuci, mboicu, dmarcu, dschum }@gmu.edu

[2] Department of Defense, USA

Abstract. This paper presents the TIACRITIS web agent and textbook for teaching intelligence analysts the critical thinking skills needed to perform evidence-based reasoning. They are based on a computational theory which views Intelligence Analysis as ceaseless discovery of evidence, hypotheses, and arguments, in a complex world that is changing all the time. TIACRITIS helps students learn about the properties, uses, and marshaling of evidence upon which all analyses rest, through regular practice involving analyses of evidence in both hypothetical and real situations.

Keywords: intelligence analysis, cognitive assistants, education and training

1 Introduction

The purpose of Intelligence Analysis is to answer complex questions arising in the decision-making process. Complex arguments, requiring both *imaginative* and *critical reasoning*, are necessary in order to establish and defend the *relevance*, *believability*, and *inferential force* of evidence with respect to the questions asked. The answers are necessarily probabilistic in nature because our evidence is always *incomplete*, usually *inconclusive*, frequently *ambiguous*, commonly *dissonant*, and with various degrees of *believability* [1, 2]. Moreover, the analysts are often required to answer questions very quickly, with insufficient time for extensive research of the available evidence.

How should the analysts be trained for such astonishingly complex tasks?

First, we think that learning to perform such complex evidential reasoning tasks cannot be done effectively just by listening to someone discuss his/her own analyses, or just by giving students lectures and assigned readings on the topics. What is absolutely necessary is *regular practice involving analyses of evidence* using either hypothetical situations or examples drawn from actual situations. In short, evidential analysis is mastered best by performing analyses contrived to illustrate the wide variety of subtleties or complexities so often encountered in actual evidential analyses. Second, based on our inspection of the materials offered in several courses for training intelligence analysts, it appears that analysts are so often trained in the

production of intelligence analyses (i.e., how to write analysis reports) rather than upon the actual *process of analysis* itself. Very little training is offered regarding the properties, uses, discovery, and marshaling of the evidence upon which all analyses rest. Our third conclusion is based on the strong emphasis currently placed in the Intelligence Community on the development of structured analytic methods and computer-based tools to assist analysts. These tools, however, to be really useful, need to have solid theoretical foundations grounded in the *Science of Evidence*.

To address the above issues, we are developing a *Computational Theory of Intelligence Analysis* which is briefly introduced in Section 2. This theory is at the basis of a textbook and web-based system, called TIACRITIS, for teaching intelligence analysts the critical thinking skills required to perform evidence-based reasoning, through a hands-on, learning by doing approach. The textbook is briefly introduced in Sections 3. Section 4 illustrates the use of the TIACRITIS system.

2 Computational Theory of Intelligence Analysis

We are developing a *Computational Theory of Intelligence Analysis*, grounded in the Science of Evidence, Artificial Intelligence, Logic, and Probabilities, to be used as a basis for building advanced cognitive assistants that:

- Support intelligence analysts in coping with the astonishing complexity of providing accurate explanations and predictions in a non-stationary world;
- Help intelligence analysts learn critical thinking skills for evidence-based reasoning through an effective hands-on approach.

As illustrated in Fig. 1, we view intelligence analysis as a process of ceaseless discovery in a non-stationary world involving mixtures of *abductive*, *deductive*, and *inductive* reasoning for *evidence in search of hypotheses*, *hypotheses in search of evidence*, and *evidential tests of hypotheses,* all going on at the same time. By means of abductive reasoning we generate hypotheses from evidence we gather; by deductive reasoning we make use of our hypotheses to generate new lines of inquiry and evidence; and by inductive reasoning we test hypotheses on the basis of the evidence we are discovering. Such testing depends on the relevance and believability of our evidence. These factors combine in further complex ways to allow us to assess the inferential force or weight of the evidence we are considering [3]. Based on this computational theory we have developed cognitive assistants for intelligence analysts that synergistically integrate three complex capabilities. They can rapidly *learn* the analytic expertise which currently takes years to establish, is lost when analysts sepa-

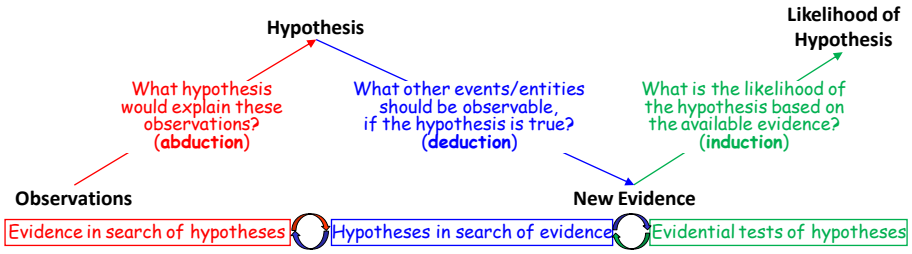


Fig. 1. Intelligence Analysis as discovery of evidence, hypotheses, and arguments.

rate from service, and is costly to replace. They can *tutor* new intelligence analysts how to systematically analyze complex hypotheses. Finally, they can *assist* the analysts in evaluating the likelihood of hypotheses by developing Wigmorean probabilistic inference networks [4] that link evidence to hypotheses in argumentation structures that establish the relevance, believability and inferential force or weight of evidence. A first prototype of such an agent is Disciple-LTA [5, 6] which is at the basis of the TIACRITIS system introduced in this paper.

3 Intelligence Analysis Textbook

The TIACRITIS textbook has been written for persons throughout the Intelligence Community and those it serves, including collectors of intelligence information, evaluators of incoming intelligence information at various levels and in different offices, and even policy-making "customers" of intelligence analysts. The matters discussed are applicable regardless of the subject of an intelligence analysis and the kinds of intelligence information required, such as HUMINT, IMINT, SIGINT, MASINT, and Open Source information.

The textbook teaches basic knowledge about the properties, uses, and marshaling of evidence to show students what is involved in assessing the relevance, believability, and inferential force credentials of evidence. It includes a wide array of examples of the use of the TIACRITIS system and hands on exercises involving both real and hypothetical cases chosen to help students recognize and evaluate many of the complex elements of the analyses they are learning to perform. Each chapter starts with a presentation of some important matter, such as assessing the believability of evidence. Then the students are asked to use TIACRITIS and experiment with what they have just been taught. Both the textbook and TIACRITIS are easily customizable by selecting the chapters and the case studies to be used.

Also discussed in the textbook is how the intelligence analysis concepts and methods embedded into TIACRITIS (e.g., the systematic approach to the development of argumentation structures, the ontology of evidence and the associated procedures for assessing the believability of evidence, the drill-down analysis and assumptions-based reasoning) help analysts perform better analyses, no matter what analysis methods they use. In particular, it shows how the very popular Richards J. Heuer's Analysis of Competing Hypothesis (ACH) method [7] can be improved by employing the concepts and methods embedded into TIACRITIS.

4 The TIACRITIS System

TIACRITIS is a web-based system with knowledge bases and case studies incorporating a significant amount of knowledge about evidence, its properties, uses, and discovery. Each knowledge base includes an ontology that defines both general concepts for evidence-based reasoning [8], and domain-specific concepts from an application domain. It also includes learned problem reduction rules and solution synthesis rules which are represented with the concepts from the ontology. These

knowledge bases allow TIACRITIS to automatically generate argumentation structures for hypotheses testing, as illustrated in Fig. 2.

The case studies are designed to learn about and practice with one new important matter at a time, such as analyzing hypotheses through reduction and synthesis, making assessments and assumptions in arguments, assessing the believability of evidence, analyzing competing hypotheses, etc. To provide an intuitive understanding of the use of TIACRITIS we present the case study "Hypothesis analysis and evidence search" which helps students learn how to search for relevant evidence on the Internet. This case study also guides the student to practice with many of the matters introduced in the previous ones. As shown at the bottom of Fig. 2, the student is instructed to select a hypothesis analysis problem and browse its analysis tree to see how it is reduced to simpler hypotheses that have to be assessed by searching for evidence on the Internet. The student will then define search criteria for the elementary hypotheses, will invoke specific search engines with those criteria, copy relevant information into TIACRITIS, define evidence from this information, associate evidence with the corresponding hypotheses, and evaluate its relevance and believability, with the goal of assessing the likelihood of the top level hypothesis.

The student is first instructed to select the **Hypothesis** menu at the top of the window. As a result, TIACRITIS will display a list of hypotheses to select from, including the option to define a new hypothesis. Next the student is instructed to select the hypothesis analysis problem *"Assess whether United States will be the glo-*

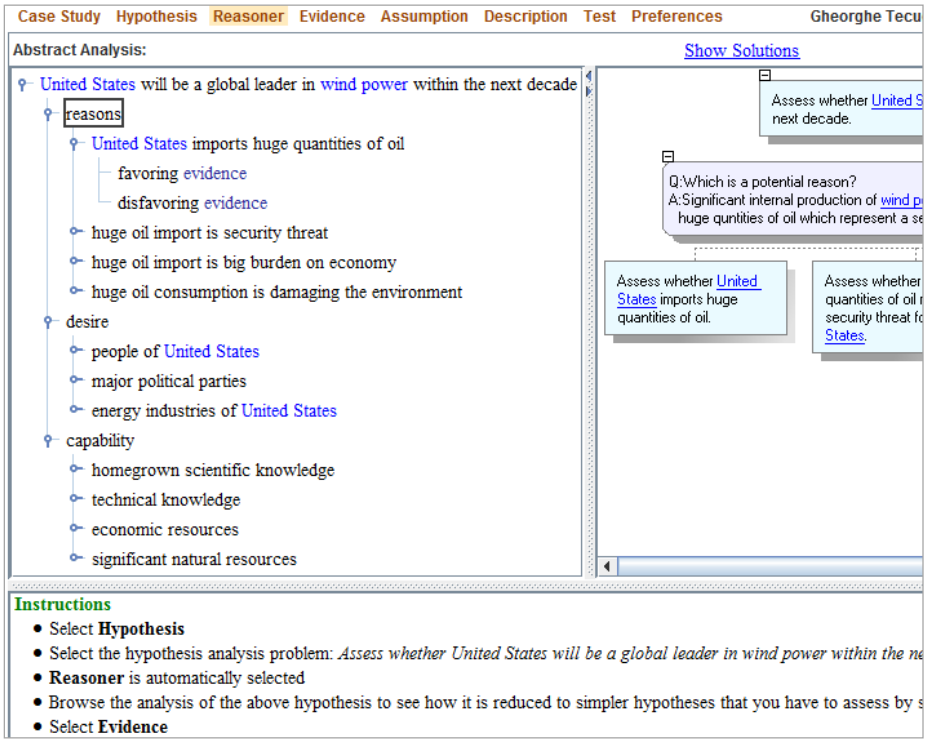


Fig. 2. The interface of the Reasoner module.

bal leader in wind power within the next decade." As a result, the **Reasoner** module is automatically invoked, generating and displaying the analysis tree from Fig. 2. Notice that the left panel displays an abstraction of the decomposition tree where the top level hypothesis is successively decomposed into simpler and simpler hypotheses, with the simplest one (such as *"United States imports huge quantities of oil"*) to be assessed based on favoring and disfavoring evidence.

The student is next instructed to browse this analysis tree. As she clicks on an abstract hypothesis in the left panel (e.g. *"reasons"*), the right panel displays the detailed decomposition of the corresponding hypothesis analysis problem, as illustrated in the right side of Fig. 2 and in Fig. 3.

Fig. 3. Detailed reasoning step.

Next the student is instructed to select the **Evidence** menu (see the top of Fig. 2) and is explained the operation modes shown in the upper part of the left panel in Fig. 4. Since **[COLLECTION GUIDANCE]** is selected, the left panel shows the elementary hypotheses and their evidential support. When the student clicks on one such hypothesis, the right panel provides more details about it. In also allows the student to associate search criteria with the selected hypothesis. For example, in the situation illustrated in Fig. 4, the student has selected the hypothesis "*Unites States imports huge quantities of oil,*" and has associated two search criteria with it, the second one being currently selected. Clicking on one of the available search engines (i.e., BING, GOOGLE, YAHOO) will invoke it with the selected search criterion. The student will use these capabilities to associate search criteria with elementary hypotheses and to search for relevant evidence on the Internet. In this example the student has identified a relevant article by Daniel Workman. She is instructed to copy it into TIACRITIS, extract items of evidence from it, specify the types of these items of evidence, and associate them with the corresponding hypotheses.

The right panel in Fig. 5 shows the defined characteristics of such an item of evidence, *EVD-001-US-top-oil-importer*. Notice its description and the item of information from which it was extracted, *INFO-001-US-oil-import*, which is the entire article. The student was then instructed to select its type from a comprehensive list of possible types. Since she selected *"unequivocal testimonial evidence obtained at second hand"*, she was prompted to specify both the name of the source of the testimony (i.e. *Daniel Workman*), and that of the primary source (*US Energy Information Administration*).

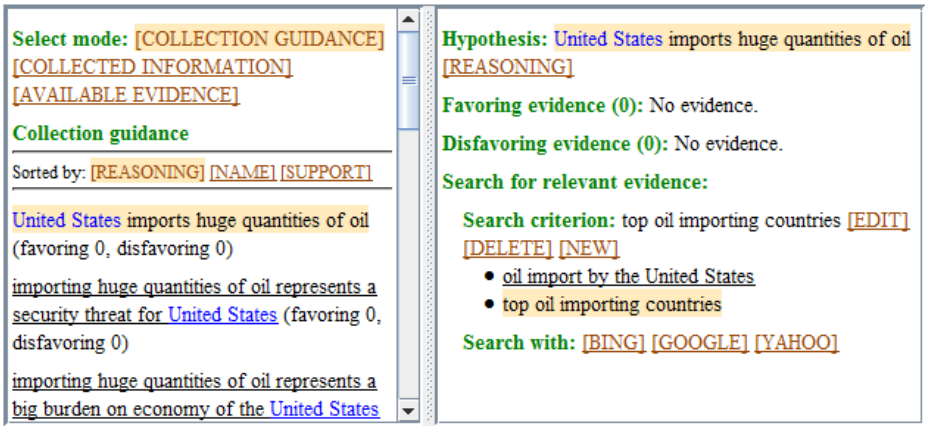


Fig. 4. Defining search criteria for a given hypothesis.

The bottom part of the right panel displays the list of all the elementary hypotheses from the analysis tree, under the label "**Irrelevant to**," each followed by four commands: [FAVORS], [DISVAFORS], [REASONING], [COLLECTION]. The student may select one of the first two commands to indicate that the current item of evidence (i.e., *EVD-001-US-top-oil-importer*) favors or disfavors that hypothesis. In the illustration from Fig. 5, the student indicated that this item of evidence favors the hypothesis that the "*United States imports huge quantities of oil.*" As a result, the "***Favors***" label was created and this hypothesis was moved under it.

The student is next instructed to select the [REASONING] command following the above hypothesis. As a result, the **Reasoner** module is invoked with "*favoring evidence*" for that hypothesis selected, as shown in the left hand side of Fig. 6. Notice that TIACRITIS has automatically generated the reasoning tree for the assessment of the relevance and believability of *EVD-001-US-top-oil-importer* with respect to the

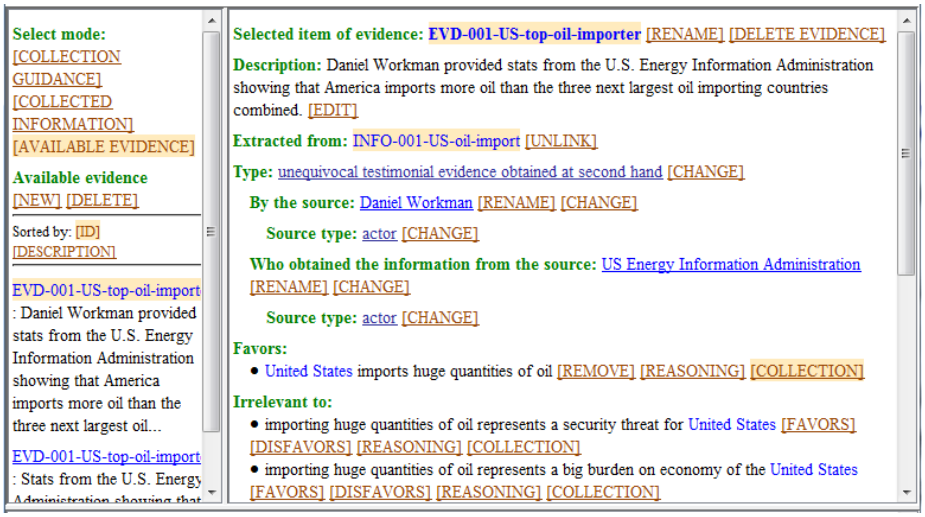


Fig. 5. Defined item of evidence favoring a hypothesis.

hypothesis that the "*United States imports huge quantities of oil.*" In particular, being *testimonial evidence obtained at second hand*, the *believability* of this item of evidence depends on the believability of its primary and secondary sources which, in turn, depend on their *competence* and *credibility*. Competence involves *access* and *understandability* while credibility involves *veracity*, *objectivity* and *observational sensitivity* [2]. The student may either assess these lower level believability credentials, or she may assess upper level ones, or even make a holistic assessment of the believability of the item of evidence. In the illustration of this case study, she clicks on "believability EVD-001-US-top-oil-importer" and then selects the **Assumptions** menu, to specify the believability of this item of evidence as an assumption. As illustrated in the right hand side of Fig. 6, TIACRITIS displays the assessment problem to be solved and a pattern for its solution. What the student needs to do is to select a likelihood, such as "almost certain" or "likely," from the menu list.

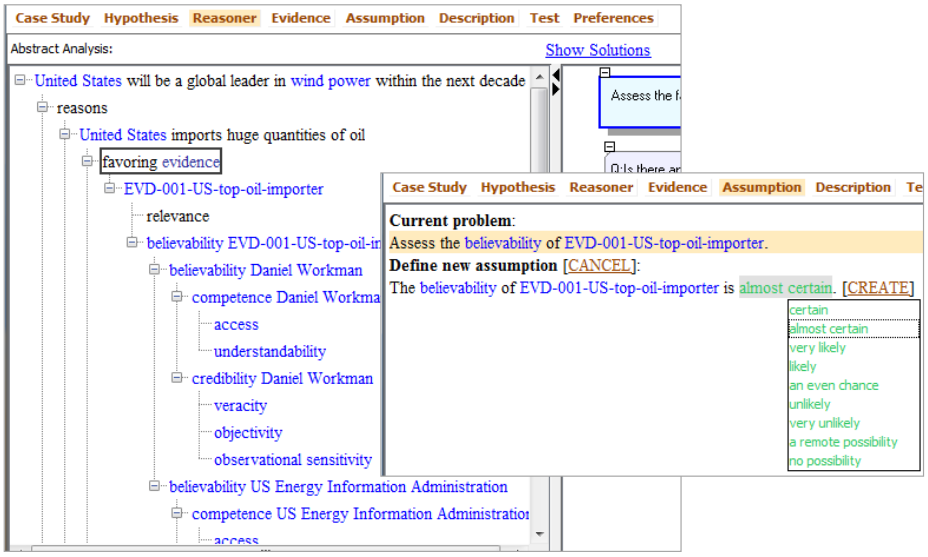


Fig. 6. Evidence assessment.

The relevance is assessed in a similar way. Both assessments are shown in Fig. 7 with a yellow background to indicate that they have been specified as assumptions. Notice also that TIACRITIS has automatically computed the inferential force of this item of evidence on some of the upper level hypotheses.

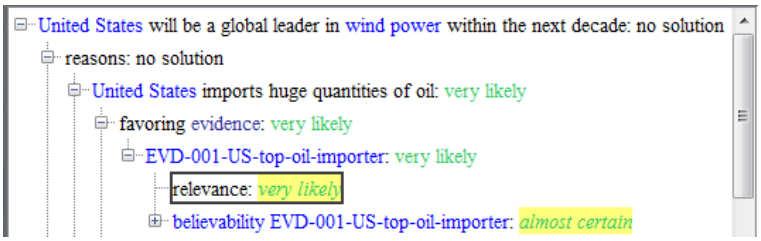


Fig. 7. Computation of the inferential force of evidence through solution synthesis.

After being guided to perform these operations, the student is instructed to complete the analyses of the top level hypothesis. Then she is instructed to define and solve a new hypothesis analysis problem, such as *"Assess whether China will be the global leader in solar power within the next decade."*

6 Conclusions

We have presented an intelligent web-based agent and an associated textbook for teaching intelligence analysts through an effective learning by doing approach. Both of them may be used as such or may be easily extended or customized with additional topics and case studies to better serve specific audiences. We plan to continuously increase the training and operational effectiveness of TIACRITIS.

Acknowledgments. We are very grateful to Phil Hwang, Don Kerr, Joan McIntyre, Kelcy Allwein, Keith Anthony, Cindy Ayers, Susan Durham, Sharon Hamilton, Jim Homer, David Luginbuhl, Bill Nolte and George Stemler for their suggestions and support. Cristina Boicu has contributed to the development of TIACRITIS. Gary Roemmick and Ben Wible have contributed to its transition to the Joint Forces Staff College and to the development of specialized case studies. This research was performed in the Learning Agents Center and was partially supported by several agencies of the U.S. Government, including the National Geospatial-Intelligence Agency, the Department of Defense, and the National Science Foundation. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and opinions expressed in this article are those of the authors and do not necessarily reflect the official policy or position of any agency of the U.S. Government.

References

1. Schum, D.A.: Evidence and Inference for the Intelligence Analyst. University Press of America, Lanham, MD (1987)
2. Schum, D.A.: The Evidential Foundations of Probabilistic Reasoning. Northwestern University Press (2001)
3. Tecuci, G., Schum, D.A., Boicu, M., Marcu, D., Hamilton, B.: Intelligence Analysis as Agent-Assisted Discovery of Evidence, Hypotheses and Arguments. In: Phillips-Wren, G., Jain, L.C., Nakamatsu, K., Howlett, R.J. (eds.) Advances in Intell. Decision Technologies, SIST 4, pp. 1--10. Springer-Verlag Berlin Heidelberg (2010)
4. Wigmore J.H.: The Science of Judicial Proof. Little, Brown & Co. Boston, MA (1937)
5. Tecuci, G., Boicu, M., Marcu, D., Boicu, C., Barbulescu, M., Ayers, C., Cammons, D.: Cognitive Assistants for Analysts. J. Intell. Community Res. Dev. (2007)
6. Tecuci, G., Boicu, M., Marcu, D., Boicu, C., Barbulescu, M.: Disciple-LTA: Learning, Tutoring and Analytic Assistance. J. Intell. Community Res. Dev. (2008)
7. Heuer, R.J.: Psychology of Intelligence Analysis, Center for the Study of Intelligence, Central Intelligence Agency, Washington, DC (1999)
8. Schum, D., Tecuci, G., Boicu, M., Marcu, D.: Substance-Blind Classification of Evidence for Intelligence Analysis. In: Ontology for the Intelligence Community, Fairfax, VA (2009)

Toward an Ontology Architecture for Cyber-Security Standards

Mary C. Parmelee

The MITRE Corporation
7515 Colshire Drive,
McLean, VA 22102-7539, USA
mparmelee@mitre.org

Abstract. The rapid growth in magnitude and complexity of cyber-security information and event management (CSIEM) has ignited a trend toward security automation and information exchange standards. Making Security Measurable (MSM) references a collection of open community standards for the common enumeration, expression and reporting of cyber-security-related information. While MSM-related standards are valuable for enabling security automation; insufficient vocabulary management and data interoperability methods as well as domain complexity that exceeds current representation capabilities impedes the adoption of these important standards. This paper describes an Agile, ontology architecture-based approach for improving the ability to represent, manage, and implement MSM-related standards. Initial cross-standard analysis revealed enough common concepts to warrant four ontologies that are reusable across standards. This reuse will simplify standards-based data interoperability. Further, early prototyping enabled us to streamline vocabulary management processes and demonstrate the ability to represent complex domain semantics in OWL ontologies.

Keywords: cyber-security, ontology architecture, security standards, security automation, making security measurable, security information and event management, SIEM, semantic interoperability, Agile Development, OWL, RDF

Disclaimer. The views expressed in this chapter are those of the author's alone and do not reflect the official policy or position of The MITRE Corporation or any other company or individual.

1 Introduction

Through its Making Security Measurable [13] and related efforts to standardize the expression and reporting of cyber-security-related information, MITRE leads the development of several open community standards. These standards are primarily designed to support security automation and information interoperability, as well as facilitate human security analysis across much of the cyber-security information and

event management (CSIEM) lifecycle. Some of the major security-related activities supported by the standards are: vulnerability management, intrusion detection, asset management, configuration guidance, incident management and threat analysis. MITRE's support of the individual standards is funded by several federal government organizations. Many of the MSM-related standards have been adopted by the National Institute of Standards and Technology's (NIST's) Security Content Automation Protocol (SCAP) program [16]. Federal government organizations and security tool vendors are moving toward adoption of SCAP validated products to ensure baseline security data and tool interoperability [15].

While MSM-related standards are valuable for enabling security automation; insufficient vocabulary management and data interoperability methods as well as domain complexity that exceeds current representation capabilities impedes the adoption of these important standards. This paper describes an Agile Development [1], ontology architecture-based approach for improving the ability to represent, manage, and implement MSM-related standards. The Cyber-Security Ontology Architecture is a loosely-coupled, modular representation that is resilient to rapid change and complexity. Architecture-based services and applications are free to combine and extend architecture components at implementation time to fit application-specific contexts without having to implement a single monolithic model. The result is improved ability to support security automation, vocabulary management, and data interoperability. Initial cross-standard analysis revealed enough common concepts to warrant four ontologies that are reusable across standards. This reuse is one way that this approach will simplify standards-based data interoperability. Further, early prototyping enabled us to streamline vocabulary management processes and demonstrate the ability to represent complex domain semantics in OWL ontologies that are difficult or not possible to represent using the Relational Database (RDB) and XML Schema (XSD) [17, 30] technologies in which the standards are currently implemented.

2 Background

This section provides background descriptions of ontology architecture and controlled vocabulary in the context of this paper.

An ontology architecture is a conceptual information model comprised of a loosely-coupled federation of modular ontologies that form the structural and semantic framework of an information domain. Ontology architectures have been used to relate upper ontologies to their middle and domain level extensions [21]. Many of the concepts involved in ontology architecture are defined Ontology architectures are especially useful when applied to large, dynamic, complex domains such as cyber-security [17]. The major benefits of this federated approach to ontology application are [8, 23]:

1. Loose coupling and modularization makes it easier to add, remove and maintain individual ontologies;
2. Modular ontologies are easier to reuse and process than large monolithic ontologies;
3. Component ontologies can be dynamically combined on demand at implementation time to meet application-specific needs.

The vocabulary of complex, dynamic domains such as cyber-security often include atypical linguistic expressions such as acronyms, idioms, and numeric codes. It is important to recognize that although these linguistic expressions are not standard language terms, they form an accepted vocabulary in the context of the domain. This perspective of what constitutes a vocabulary calls for a broad definition of controlled vocabulary (CV). In this context, a controlled vocabulary is a collection of linguistic expressions that is vetted by an authority (e.g. a community) according to a set of criteria. All of the MSM standards maintain some form of a controlled vocabulary. These vocabularies were developed independently of each other, and are at various stages of maturity that range from a few months to ten years of active development.

3 Obstacles to Standards Adoption

The three major obstacles inhibiting the widespread adoption of the MSM-related standards are:

1. Unsustainable vocabulary management processes: Vocabulary management involves thousands of manually developed and managed value enumerations and vocabulary representations that are mostly encoded in XSD. The MSM-related standards are growing rapidly in number, volume and complexity. Some of the standards are adding hundreds to thousands of enumeration entries per month. A semantic approach to vocabulary management would streamline the vocabulary management process and reduce human error.

2. Ineffective data interoperability methods**:** Data interoperability activities are largely driven by the SCAP Validation program, which among other things, requires security tool vendors to translate proprietary output to a common expression and reporting form in order to achieve SCAP compliance [15]. This data interoperability is typically accomplished with manual ETL-style mappings to each of the SCAP-required standards. This mapping process would be more tractable, even semi-automatable if common concepts were represented more consistently across standards. A well-designed ontology architecture would facilitate this consistency.

3. Rapidly evolving, complex domain semantics that exceed the representation capability of the RDB and XSD technologies in which the standards are currently implemented: Domain complexity issues such as how to represent the behavioral

aspects of malware, and relating numerous software versioning schemes, call for a more semantic representation than either XSD or RDB technologies alone can readily provide. The semantics of these technologies are currently represented mostly in human interpretable documentation, which is not automatable or machine processable.

The following sections of this document describe how a well-designed ontology architecture coupled with a semantic technology-based approach to information management could improve the productivity and efficiency of MSM-related standards development, management and implementation [19, 20].

4 Agile Development Approach

We take an Agile Development approach (Agile approach), to ontology architecture design, development, and implementation [1]. Agile Development begins with an envisioning phase in which we rapidly collect and prioritize user needs, perform coarse grained architecture modeling, and roughly estimate scope. Then we implement the architecture by building incremental capability in short design and development cycles called sprints. The intent is to allow the architecture to gradually evolve based on emerging stakeholder requirements and lessons learned from each sprint [1]. When fully mature, the Cyber-Security Ontology Architecture will represent a comprehensive, standards-based family of ontologies.

4.1 Envisioning Phase

We gathered high level requirements from domain experts, which are expressed as obstacles to adoption in Section 3 of this document. Then we developed a coarse model of the CSIEM lifecycle to provide a rough estimate of scope. We mapped the current MSM-related controlled vocabularies (CVs) to the CSIEM lifecycle model to produce a CV architecture as illustrated in Figure 1. Acronym expansions for the standard names in Figure 1 are located in the References section, reference numbers 1,2,3,4,5,6,7,12,14,18, and 29.

Finally, we performed a vocabulary analysis, identifying gaps and overlaps while extracting common concepts for reuse across vocabularies. Results are illustrated in the first draft Cyber-Security Ontology Architecture as illustrated in Figure 2 [2,4,6,18]. The top two layers of the architecture designates the ontology-level tiers. We will eventually fill the gaps with new or existing ontologies while reducing vocabulary overlap to only intentional variation in order to control complexity and improve structural and syntactic information interoperability.

The lowest tier of the architecture designates the standards value-level CV content followed by the CV representations in the third tier. These two CV tiers are the sources for the upper two ontology-level architecture tiers. Above the CV tiers, the

third tier contains ontologies that are specific to the cyber-security domain. Finally the upper-most tier contains common ontologies that emerged from vocabulary overlap analysis. Development of the first draft Cyber-Security Ontology Architecture marks the end of the envisioning phase of development and the beginning of Sprint 1 implementation. The ontologies that are encircled with red ovals are those that have been developed or adopted during the Sprint 1 implementation phase. We adopt or derive from existing ontologies where possible. The ontologies are encoded in the Web Ontology Language (OWL) [25].

Fig. 1. CSIEM CV Architecture

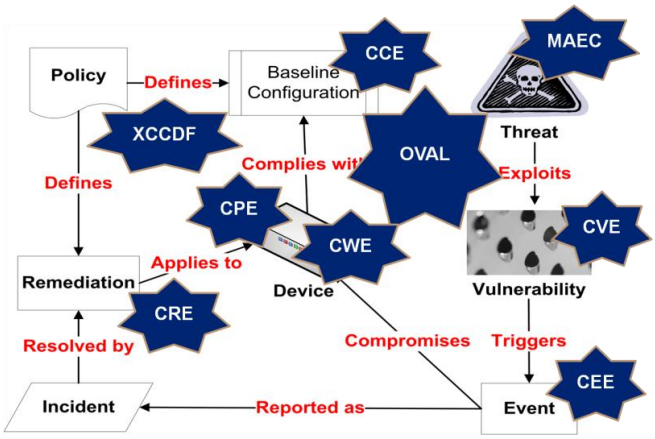
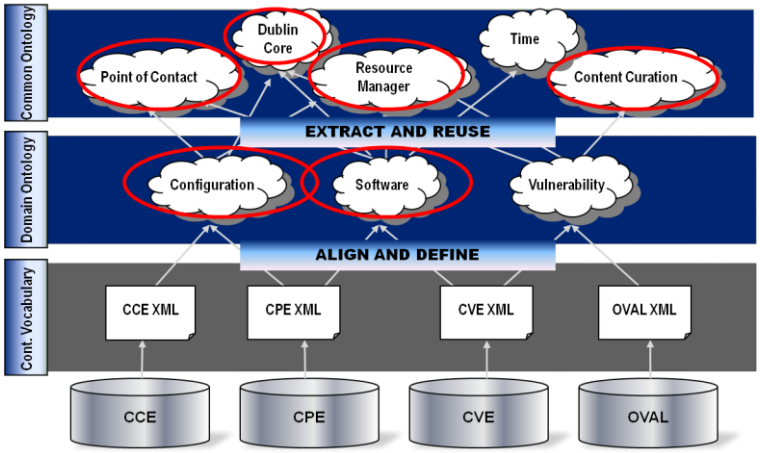


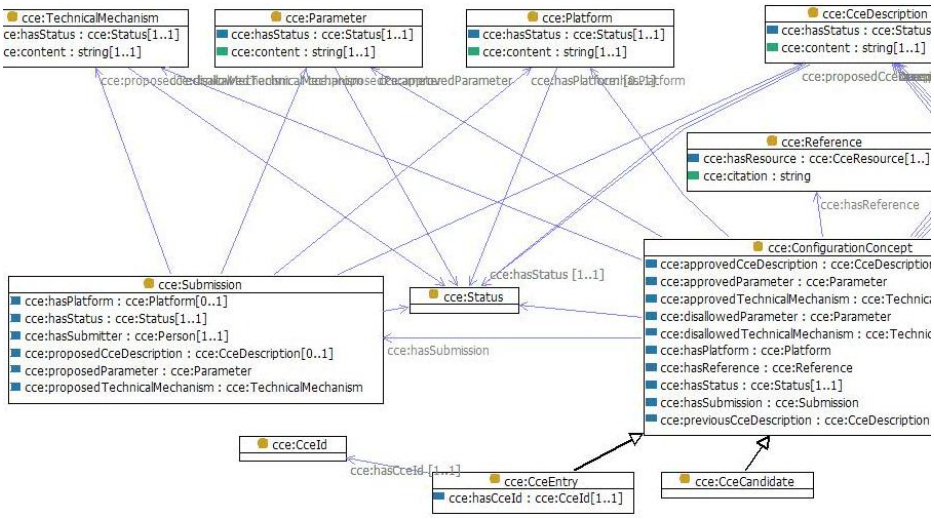
Fig. 2. Cyber-Security Ontology Architecture Concept Diagram



4.2 Cyber-Security Ontology Architecture Implementation Sprint 1

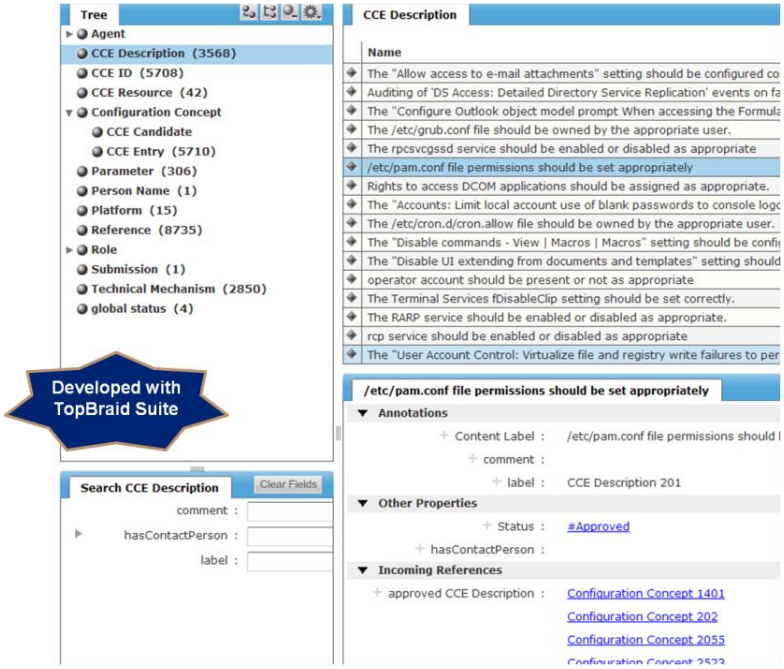
Sprint 1 focused on improving the vocabulary management process and produced five ontologies. Four of these are common ontologies, including: an OWL (Web Ontology Language) representation of the Dublin Core metadata standard [9,25]; a Resource Manager ontology which imports the Dublin Core model and references parts of SKOS (Simple Knowledge Organization System) [28]; a Point-of-Contact ontology (which was derived from the FOAF [10] and VCard ontologies) [26]; and a Content Curation ontology. The domain ontology was derived from the Common Configuration Enumeration (CCE) CV. It includes the Content Curation ontology and parts of the other three common ontologies. Figure 3 illustrates the structure of the CCE Vocabulary Manager Ontology's core concepts.

Fig. 3. CCE Vocabulary Manager Ontology Core Concepts



We converted the existing CCE XML content into over 27,000 RDF [27] instances to create the CCE Vocabulary Manager knowledge base, which contains over 500,000 RDF triples. Then we implemented a reference Semantic Web application using Top Quadrant's TopBraid Suite [24]. This application enables CCE content analysts to view, query, navigate, edit and track the status of CCE content in the knowledge base. Figure 4 shows a screenshot of the CCE vocabulary management application. The RDF graph structure eliminates the need for redundant content that is required of tabular and hierarchical structures. The OWL ontology expands the single tacit CCE Entry relation to many explicit user-defined relations among CCE instances. These capabilities, among others, have the potential to streamline vocabulary management processes and improve content quality across MSM-related standards.

Fig. 4. CCE Vocabulary Management Web Application



5 Future Work

In the near future, we will refine the vocabulary management reference application while building out the ontology architecture. A longer term goal is to develop an end user reference implementation that semi-automates the mapping of proprietary tool output to standard vocabularies.

Acknowledgement Thank you to MITRE subject matter experts Matthew Wojcik, Jonathan Baker and David Mann for their valuable contributions to this research.

References

1. Amber, Scott W.: Agile Model Driven Development, http://www.agilemodeling.com/essays/amdd.htm
2. CCE: Common Configuration Enumeration, http://cce.mitre.org/
3. CEE Board: Common Event Expression Technical Report, Department G026, The MITRE Corporation (2007)
4. CPE: Common Platform Enumeration, http://cpe.mitre.org/files/cpe-specification_2.2.pdf

5. CRE: Common Remediation Enumeration, http://scap.nist.gov/events/2010/saddw/presentations/remediation.pdf
6. CVE: Common Vulnerability and Exposures, http://cve.mitre.org/
7. CWE: Common Weakness Enumeration, http://cwe.mitre.org/
8. Deshayes, Laurent; Foufou, Sebti; et al.: An Ontology Architecture for Standards Integration and Conformance in Manufacturing, 6th International IDDME, Grenoble, France, May 17-19 2006. http://stl.mie.utoronto.ca/publications/P0057paper.pdf
9. Dublin Core Metadata Inititative: Dublin Core Element Set, http://dublincore.org/documents/dces/
10. FOAF: Friend-of-a-Friend Vocabulary Specification, http://xmlns.com/foaf/spec/
11. ISO: ISO 639-4:2010, http://www.iso.org/iso/catalogue_detail.htm?csnumber=39535
12. MAEC: Malware Attribute Enumeration and Characterization, http://maec.mitre.org/
13. MSM: Making Security Measurable, http://measurablesecurity.mitre.org/
14. Mann, David: An Introduction to the Common Configuration Enumeration (CCE), Technical Report, Department G022, The MITRE Corporation (2008)
15. NIST: Interagency Report 7511, SCAP Validation Derived Test Requirements, http://csrc.nist.gov/publications/drafts/nistir-7511/draft-nistir-7511_rev1.pdf (2009)
16. NIST: SCAP (Security Content Automation Protocol), http://scap.nist.gov/
17. Obrst, Leo: *Ontological Architectures*, *Chapter 2 in Part One: Ontology as Technology* in the book: TAO – Theory and Applications of Ontology, Volume 2: The Information-science Stance, Michael Healy, Achilles Kameas, Roberto Poli, eds. Springer, (2010).
18. OVAL: Open Vulnerability and Assessment Language, http://oval.mitre.org/
19. Parmelee, Mary: Toward the Semantic Interoperability of the Security Information and Event Management Lifecycle, In: AAAI Intelligent Security Workshop, http://www.tzi.de/~edelkamp/secart/IntSec.pdf (2010)
20. Parmelee, Mary; Nichols, Deborah; Obrst, Leo: A Net-Centric Metadata Framework for Service Oriented Environments. IJMSO 4 (4): 250 – 260 (2009)
21. Pease, A., Niles, I., and Li, J.: The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications. In Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web, Edmonton, Canada (2002)
22. Princeton University: WordNet, http://wordnet.princeton.edu/
23. Probst, F., M. Lutz: Giving Meaning to GI Web Service Descriptions, WSMAI (2004)
24. Top Quadrant: TopBraid Suite, http://topquadrant.com/products/TB_Suite.html
25. W3C: OWL Overview, http://www.w3.org/TR/owl-features/ (2004)
26. W3C: Representing vCard Objects in RDF, http://www.w3.org/Submission/vcard-rdf/ (2010)
27. W3C: Resource Description Framework (RDF) Semantics, W3C Recommendation http://www.w3.org/TR/rdf-mt/ (2004)
28. W3C SWDWG: SKOS, http://www.w3.org/2004/02/skos/ (2004)
29. XCCDF: Specification for the Extensible Configuration Checklist Description Format (XCCDF) Version 1.1.4, http://csrc.nist.gov/publications/nistir/ir7275r3/NISTIR-7275r3.pdf (2008)
30. W3C XSWG: XML Schema Part 1: Structures, http://www.w3.org/TR/2001/PR-xmlschema-1-20010330/ (2001)

Position Papers

Semantic Real Time Intelligent Decision Automation

Bill Guinn
Amdocs Inc., 1104 Investment Boulevard
El Dorado Hills, CA 95762
www.amdocs.com
bill.guinn@amdocs.com

Jans Aasman
Franz Inc., 2201 Broadway, Suite 715
Oakland, CA 94612, USA
www.franz.com
ja@franz.com

Abstract: The Intelligence Community has systems that deal with overwhelming numbers of events that have to be analyzed in real time. Ideally these systems predict malicious events and aberrant human behavior far enough in advance so that appropriate action can be taken. The challenge is that events come from various real time sources and various databases and do not fit together well. The Intelligence Community (IC) has recognized that semantic technology might be of help in a number of ways. Semantic Technology helps in integrating event streams from various sources and it can describe the people, organizations, relationships, situations and threats in a more declarative way so that it is easier to disambiguate people and groups and it is easier to write rules and analytics to generate predictions. There are a number of attempts in the intelligence community to build Semantic Platforms that can do what we just described. There is also one system that does just this in a commercial setting.

Keywords: Event Processing, Bayesian Belief Networks, CRM, Telecommunications, Semantic Web, Rule Based System, Triple Store

1. Introduction

This article describes a Semantic Platform that is used in a Telecommunications setting, which deals with the same magnitude of complexity as the IC and definitely has the same massive scaling challenge: support 10 billion events per day with sub second response time.

2. AIDA - Amdocs Intelligent Decision Automation

The AIDA Semantic Platform is built by Amdocs in cooperation with partners like Franz Inc. for the AllegroGraph triple store, Norsys Software for a Bayesian Belief Network system, and Gigaspaces Technologies for the Java middleware layer. Amdocs is a publicly traded company (NYSE: DOX) that is the market leader in customer experience systems for Telecommunications and Cable companies including Billing, CRM, Network Planning and Provisioning.

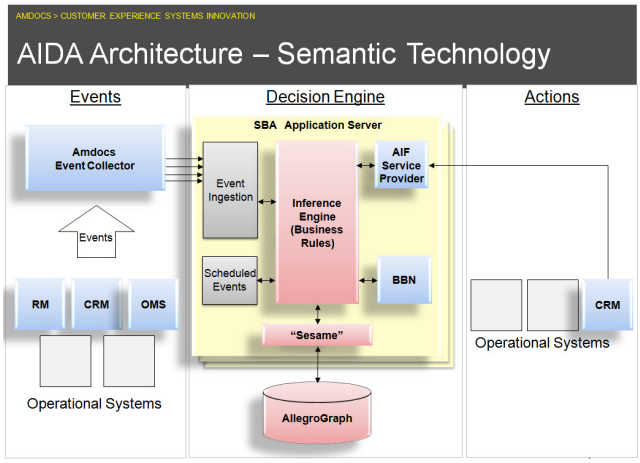
2.1 What is the platform used for?

The first use case that is currently being marketed by Amdocs is for improving customer care. Currently running a Call Center is one of the most costly elements in the total operational budget of any service oriented business. In North America the average cost of servicing a call with a Customer Service Representative (CSR) is $.50/minute. Large organizations which may have tens of thousands of CSRs, can save millions a year simply by shaving a few seconds off of their support calls. Studies showed that on average the CSR will go through 68 screens in one customer interaction. Wouldn't it be perfect that when you call, the CSR already knows what you are calling about and has the solution at hand? Both the Telecom and the customer win.

2.2 What are the elements of the platform?

Figure 1 shows some of the important elements: an Event Collector, a Decision Engine, the AllegroGraph triple store, a Bayesian Belief Network and a workbench for analysts and business people. Combined, this pipeline of

technology implements an event-condition-action framework to drive business processing in real time. We'll discuss each of the platform's components in more detail.



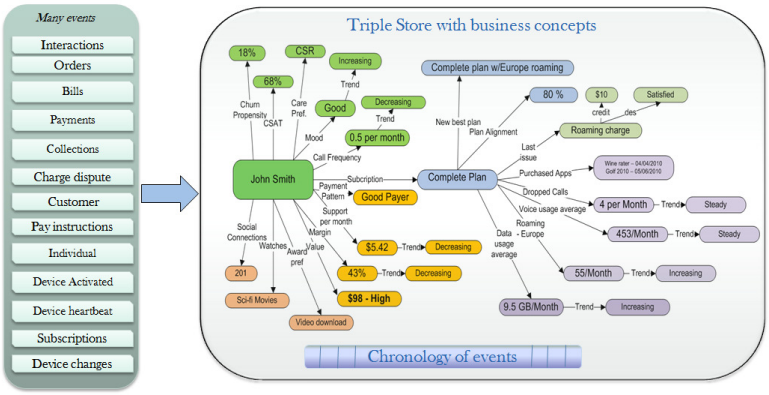
2.2.1 Event Collector

The Event Collector is a tool that is connected to all the systems that are usually involved in running the Telecom business. Examples of the data inflow are Call Detail Records, new and terminating subscriptions, outgoing bills, payments and collections, device heart beats, device changes, and network sensor data. However, there are also new sources of information: location based information, e-commerce events, sending of SMS and email, application downloads on iPhones or Android phones, etc, etc.

2.2.2 Decision Engine

The most important role of this component is [a] to compute over 100 high level business concepts that describe a customer in detail, and [b] to predict a customer behavior – as an example, when and why a customer will contact the Telco. The workflow is roughly as follows. First, the Event Collector receives a new event and encodes it as a set of RDF triples. Second, the Decision Engine then retrieves all the known triples about the subject of the event (for example a particular customer) and combines it with the new event triples and re-computes any affected high level business concepts. Part of this computation is done by invoking a Bayesian Belief Network (BBN). The BBN generates new predictions that are stored as triples. Figure 2 is an illustration of how events lead ultimately to a semantic network describing a customer. Note that figure 2 shows only a few percent of the total knowledge of a customer. Note also that this Decision Engine contains a rule based system that was developed by Amdocs. Due to their non-deterministic performance, RDFS or OWL reasoning was not used. Instead a custom forward and backward chaining rules engine was built to execute hundreds of inferences in milliseconds.

Events from many source systems are transformed into a set of related business concepts



2.2.3 Bayesian Belief Network

An important part of the platform is to predict behavior. Typical inferences are:

- Will the customer make a payment on time?
- Is there a reason (such as usage on a customer's bill) which will motivate the customer to call support?
- When is the customer likely to cancel service?

In other words the system predicts the likelihood of something happening estimated by the frequency of particular configurations of events and attributes. This can apply to a single individual or for a group of individuals that are roughly the same (for some statistical and semantic interpretation of 'same'). Just using ontologies and events represented as RDF is not sufficient to compute these likelihoods. One needs statistical and machine learning techniques combined with the knowledge expressed as triples to generate predictions.

BBNs provide a natural method for representing probabilistic dependencies among a set of variables and events. The BBN used in Amdocs' platform is an integral part of the Decision Engine.

2.2.4 Triple Store (AllegroGraph)

AllegroGraph fulfills a number of requirements that make it useful as an event database. Actually, a number of customers, even within DOD, use AllegroGraph in this capacity. Let us look at a simplified ontology of an event. We see that an event has the following:

- Hierarchy of types - Think of meetings, communications event, financial transactions, visit, attack/truce, an insurance claim, a purchase order, an observation. Working with types requires RDFS++ reasoning
- List of actors - Any of examples from the previous paragraph involve at least two actors, but usually more. In order to establish social networks, distances and strengths of connections between people and to find leaders in groups requires Social Network Analysis (SNA) techniques. AllegroGraph has an extensive SNA library.
- A place - Nearly all of the events above happen somewhere. AllegroGraph provides extremely efficient Geospatial indexing and GeoSpatial operators like proximity search.
- A time - All of the above events have a start time and possibly an end time or duration. AllegroGraph also provides efficient Temporal Indexing and Temporal Reasoning.

AllegroGraph does all of the above and works with billions of triples per machine instance. However, Amdocs takes AllegroGraph to a complete new level in terms of scalability. In Amdocs' AIDA Platform, all customer and device related events and even the predictions are stored as RDF triples in AllegroGraph. The scalability challenges are non-trivial. Consider a Telecom company with 80,000,000 customers. The typical number of events seen throughout the enterprise will total around 8 billion a day. Assuming each event generates on average 20 triples, you have generated a trillion triples in a few days.

Here are some of the new requirements for AllegroGraph in this platform:

- The triple store allows for a constant insert speed at more than 100,000 triples per second. All the triples need to remain indexed at all time and queries are interspersed with additions and deletions.
- The triple is transactional, even at high speed and high scale, and needs to be replicated at all times for scalability and high availability.

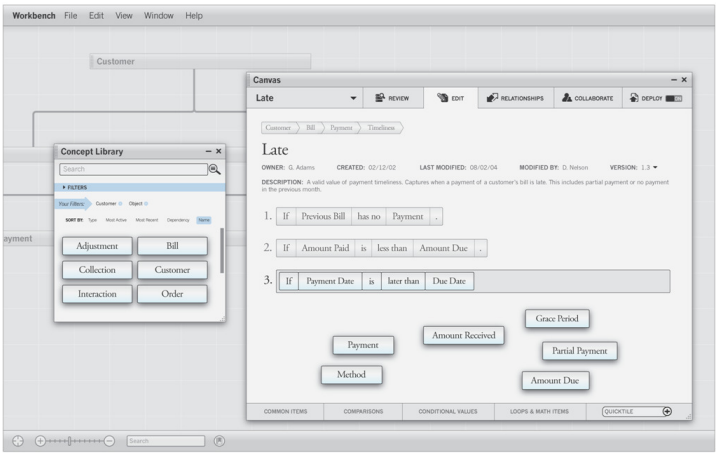
Currently AllegroGraph 4.0 can reach this scale by carefully partitioning the event data over multiple instances. However, in order to make this partitioning easier Franz is working on customizations that will offer High Availability that is completely distributed and can transparently deal with trillions of triples.

2.3 Actions

Once we have seen what is transpiring through the capture of an event, thought about what it means through inferencing and updating over 100 business concepts, we need to take the appropriate action. Using standard SOA integration schemes, the platform invokes the appropriate applications to drive the customer experience, and overall business processes. With this complete event-condition-action capability, a closed loop is created where the effect of our actions are seen in turn as new incoming events. This allows us to create goal oriented self-optimizing processes, and to learn from the actions taken under specific circumstances using backward chaining rules. This inherent feedback loop enables training and refining the BBN for better predictions along with honing the policy model which controls the business.

2.4 Workbench

A critical requirement for the AIDA Platform was that policies, rules, and the declaration of the ontology could be composed by analysts and business people who can't write code, and have no idea how to define an ontology, train a BBN, or write SPARQL. So Amdocs implemented a Workbench on top of their rule and decision engine which controls the declaration and configuration of the entire event-condition-action scheme. Figure 3 demonstrates a business user interface for a sophisticated "Magnetic Poetry", essentially a GUI to create if-then rules and policies. These if-then rules are translated into Java code (Figure 4) and SPARQL. When these rules are executed, new business concepts are generated and stored again as triples in the triple store.




```
rule PaymentDetails.timeliness
{
  if date within EarlyPeriod days after customerBill.billDate
  then timeliness = Early ;
  else if date not within LatePeriod days after customerBill.billDate
  then timeliness = Late ;
  else timeliness = OnTime ;
}
```

Each business rule defines an attribute. This rule defines an attribute of the PaymentDetails class called timeliness

All classes and their attributes are defined in the application ontology

3. Conclusion

This paper provides a brief review of a commercially available Semantic Platform for handling over 10 billion events a day in a very low latency, high availability implementation. By monitoring customer interactions and their network activities, and predicting customer behavior we can take immediate action to ensure an optimized business and customer experience. We believe that there are many similarities with the workflow and processes within the Intelligence Community that should be explored.

Using Ontologies to Mitigate LDAP Deficiencies

Joshua Powers
Securboration, Inc.
1050 W. Nasa Blvd. Suite 156
Melbourne, FL 32901
jpowers@securboration.com

Abstract. Semantic technology powered access control schemes have been recently proposed to enhance the flexibility of role-based access control (RBAC) and its variants. These access control mechanisms depend heavily on rich, contextual data sourced from an identity attribute store. Unfortunately, most identity stores in use today use the Lightweight Directory Access Protocol (LDAP) representational schema which has several deficiencies as a knowledge representation, particularly when applied to fine-grained, contextual access decision policies. This paper reviews some of these gaps and shows how the same semantic infrastructure used for the access control mechanisms can be employed to mitigate LDAP assumptions.

Keywords: access control, identity management, authorization, semantic technology, LDAP, RDF/OWL

1. Introduction

In the past decade, the defense and intelligence communities have acknowledged the importance of moving from a 'need to know' assumption to a 'need to share' assumption with respect to the secure exchange of information [1] [2]. This has been interpreted in a number of ways, including reducing barriers between networks, establishing enterprise service buses, and building metadata repositories, federated search schemes, enterprise catalogs and enterprise-level portals.

At the same time, adoption of Service Oriented Architecture (SOA) standards has made the information delivery mechanisms themselves increasingly modular and decoupled from stovepipe systems of record. Access to authoritative data about a subject of interest requires the availability of a simple endpoint, usually a URL over some standard protocol, rather than a complex point-to-point integration between two large networks or systems.

These changes have not gone unnoticed by the information assurance and security communities. Mandatory Access Control (MAC) schemes that protect data at different levels of classification are still largely in effect, although secure cross-domain technologies are attempting to break some of those sharing barriers. More importantly, within the same classification level, Discretionary Access Control

(DAC), or any variant involving assignment of individual requestor privileges to individual resources, cannot scale to a goal of ubiquitous information sharing with unanticipated but qualified requestors.

To address the sharing assumption, information assurance efforts have looked at Role-Based Access Control (RBAC) [3], a more flexible protection model initially developed for industry, and a more generic formulation called Attribute-Based Access Control (ABAC) [4]. This model's original characterization is fairly vague in terms of specifying representational mechanisms, so semantic technology approaches have been suggested for formalizing ABAC. While these access control models have advanced to keep up with new information sharing requirements, there is an unfortunate gap in the representational state of the authoritative data that provide the critical information about requestors used to decide and enforce policies under these advanced access control models. These data are most often stored and managed in Lightweight Directory Access Protocol (LDAP) directories. In this paper, we first describe a couple of semantic technology-based access control schemes and the underlying identity attributes they require. Then we show the specific technical barriers presented by LDAP in addressing these requirements. With each barrier, we show how semantic technologies similar to those used in the access control models and policies can be brought to bear to mitigate deficiencies in these attribute stores. We conclude the paper with suggestions for future work.

2. Semantic Access Control Schemes

RBAC itself does not limit the attributes associated with requestors to any particular degree of granularity, complexity or context. However, in practice, RBAC typically uses a Distinguished Name (DN) for identification purposes, plus a set of group memberships, role occupancies and basic demographic data. It does not usually account for attributes of entities which form a context around the requestor, the resource and the nature of the request.

One approach to increase the flexibility of an access control decision is the Semantic Policy Broker [5]. This mechanism causes authorizations to flow through an ontology, following its graph-like structure through an arbitrarily wide context. A natural language description of a complex policy might be:

> *"An engineer can view information about a mission which a piece of equipment that they work with supports if they are part of the organization that owns that mission."*

Under most interpretations of RBAC, this would result in a mapping of users to roles, each role representing their participation in a mission:

```
mission1_role
   roleOccupant: user1
   roleOccupant: user2
mission2_role
   roleOccupant: user1
```

. . .

All of the intermediate context involving membership in an organization, mission support, equipment, etc. is left to an administrator to work out role-by-role. Using the Semantic Policy Broker instead, an administrator translates such a policy into a SPARQL query such as:

```
(?requestor rdf:type sempbro:Person)
(?requestor sempbro:memberOf ?organization)
(?requestor sembpro:engineers ?equipment)
(?organization sempbro:owns ?mission)
(?equipment sempbro:supports ?mission)
```

This query then satisfies for some combinations of requestors and missions and does not for others.

Another approach is ROWLBAC [6], which represents the roles, requestors, resources and permission decisions of RBAC as OWL DL classes. Some attention is given to the temporal relevancy of roles, either determined by a requestor's own assertion or by some additional, higher-level rules regarding the different roles which are relevant to a given request type.

The authors of these approaches have carefully left the nature of the identity store which would support their rules with instance data out of the scope of their discussions. Organizations likely to benefit from advanced access control models such as those above are almost certain to have their identities stored and managed in an LDAP directory.

3. LDAP/LDIF

LDAP is a binary protocol for querying and modifying directory data. It also specifies the representational scheme which is used in these directories. This is serialized in readable text as LDAP Data Interchange Format (LDIF) entries of the following sort:

```
dn: cn=John Smith,ou=Users,ou=People,dc=dod,dc=mil,c=us
cn: John Smith
mail: jsmith@dod.mil
employeeid: 123456789
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
```

There are two hierarchical representations of where John Smith resides in the directory: a sort of structural class membership given by the objectclass attributes, and a sort of group membership given by the distinguished name string. Objectclasses such as person, organization, and organizationalRole are predefined by various LDAP RFCs. They determine which attributes may be used in an entry

of that type. The distinguished name reveals the hierarchy of groups, each of which is an instance of one of these various objectclasses.

LDAP is well-suited to provide rapid lookup of simple attributes to determine who may join a network, use a printer or perform other basic functions. It is not particularly useful in representing the kind of contextual information needed for the advanced access control models discussed above.

4. Compositionality

A subject's LDAP unique ID within the directory is the concatenation of an entity's group memberships in inclusion order. This presents a fragility with respect to organizational change over time which LDAP administrators have recognized. As a result, almost no interesting group membership is asserted within a typical LDAP directory outside of basic 'User,' 'Admin,' and 'Roles'. These groups are then included within a high-level group representing the entire enterprise. This approach conflicts with access control schemes whose decisions are based on finer-grained group membership information. Within the DoD, it is common practice for each Department to set up a high level LDAP group for contractors, one for civilian employees, one for reserve duty members and another for active duty members. Many DoD contractors are in fact reserve members as well. This does not mean, from an LDAP perspective, that they have two roles with respect to the same organization. It means that they are actually two different people depending on which credential they present to an access decision point. A separation of unique identification from group membership statements is a natural approach in an OWL ontology:

```
<ldap:Person rdf:ID="Person1">
  <ldap:name>John Smith</ldap:name>
  <ldap:employeeid>123456789</ldap:employeeid>
  <ldap:memberOf rdf:resource="ldap#ReportingUnit12">
...
</ldap:Person>
```

A distinguished name string may be stored explicitly as another property in the ontology or may be constructed by a traversal of membership relations if it is needed for legacy purposes.

5. Transitivity

There are two types of properties in an LDAP structure: those which range over string values and those which range over distinguished names. Neither of these property types may enforce transitivity within the directory. Outside of the group memberships that make up the distinguished name structure and the structural objectclasses, there is no support for transitive properties. This means that any role

or permission based on such a property must be 'flattened out' representationally and added one-by-one for each 'level' of entity so connected by the property.

Within the DoD, there are transitive command properties that are critical for access decision making. Administrative Control (ADCON) is the military doctrinal interpretation of Federal government management responsibilities. Operational Control (OPCON) authorized the employment of resources to accomplish assigned missions. Tactical Control (TACON) authorizes direct control of movements or maneuvers.

OWL ontologies, and the reasoners that operate on them, have built-in support for transitive properties:

```
<owl:TransitiveProperty rdf:ID="ADCON">
  <rdfs:domain rdf:resource="#MilitaryUnit"/>
  <rdfs:range rdf:resource="#MilitaryUnit"/>
</owl:TransitiveProperty>
...
<ldap:MilitaryUnit rdf:ID="ReportingUnit12">
  <ldap:name>Tech Platoon 12</ldap:name>
  <ldap:ADCON rdf:resource="#ReportingUnit34">
...
</ldap:MilitaryUnit>
```

The basic LDAP directory hierarchies, both structural and group membership, may also be represented to support legacy uses of the data. However, for the advanced access control schemes discussed above, it is only necessary to represent the properties and classes dealing with those portions of the real world needed to make the access decision.

6. Administration

In an organization the size of the DoD, or even one of its Departments, managing thousands of roles across tens of thousands of units and associating them with millions of employees is a daunting task no matter what technology is used. Choosing a representational scheme that does not allow transitive properties and that concatenates unique IDs based on membership information that may change exacerbates the administrative issues.

More concerning for administrative complexity and resource use is that detailed access decisions do need to be made. If they are not supported by the LDAP infrastructure, which is usually at enterprise or sub-enterprise level, it becomes the responsibility of individual application administrators to put requestors on access control lists (ACL) for resources, one-by-one.

An enterprise-level attribute store which has the representational power to match the fine-grained access control needs of resources housed in disparate applications will reduce redundancy of administrative effort. It should also increase the

robustness of the organization's cyber defense posture since the distributed administrative burden makes it hard for an enterprise monitor to observe the actions of a single requestor across many applications.

7. Scalability

LDAP directories can be provisioned in distributed fashion, across a number of physical servers. However, the largest LDAP implementations generally cover a few million personal accounts with a couple dozen organizational accounts and a couple dozen attributes.

The Lehigh University Benchmark (LUBM) [7] is the open test platform for RDF/OWL triple stores. Triple stores regularly handle SPARQL queries over billions of triples on fairly modest servers [8] [9].

8. Discussion and Future Work

The implementation of the data store which an LDAP-compliant server uses is not specified by the protocol. All of the "ins and outs," however, must comply with the LDAP representational schema. This admirable decoupling offers the possibility of implementing an RDF triple store as the LDAP server's database and wrapping it with fully LDAP-compliant services. This would seemingly defeat the purpose of the triple store's more useful representational schema, but it would offer the possibility of a 'side-by-side' set of RDF/OWL and SPARQL services that could be used by the advanced access control schemes discussed above. Development of such a hybrid server will be part of our future work.

The Security Assertion Markup Language (SAML) is an important recent development for communicating authorization attributes. Its XML-based format currently assumes LDAP-like contents, but could be easily extended to allow direct reference to ontology assertions.

9. References

[1] J. X. Dempsey, "Moving from 'Need to Know' to 'Need to Share:' A Review of the 9-11 Commission's Recommendations". Testimony before the House Committee on Government Reform, August 3, 2004.

[2] Office of the Director of National Intelligence, Intelligence Community Directive Number 501. http://www.dni.gov/electronic_reading_room/ICD_501.pdf.

[3] D. F. Ferraiolo, D. R. Kuhn, "Role-Based Access Controls". In Proceedings. 15th National Computer Security Conference. 1992.

[4] H. Shen, F. Hong, "An Attribute-Based Access Control Model for Web Services". In Proceedings. 7th International Conference on Parallel and Distributed Computing, Applications and Technologies. 2006.

[5] B. McQueary, A. P. Stirtzinger, “Semantic Policy Broker Final Technical Report”. Prepared for Air Force Research Labs. 2009

[6] T. Finin, A. Joshi, et. al. “ROWLBAC - Representing Role Based Access Control in OWL”. In Proceedings. ACM Symposium on Access Control Models and Technologies. 2008.

[7] Y. Guo, Z. Pan, J. Heflin. “An Evaluation of Knowledge Base Systems for Large OWL Datasets”. In Proceedings. International Semantic Web Conference. 2004

[8] Ontotext LUBM Performance Report. http://www.ontotext.com/owlim/benchmarking/lubm.html

[9] O. Erling, I. Mikhailov. “RDF Support in the Virtuoso DBMS”. Technical Report. http://www.openlinksw.com/uda/wiki/OdbcRails/main/Main/VOSArticleRDF/rdfdb1.pdf