2004

Semantic Visualization: Interfaces for Exploring and Exploiting Ontology, Knowledgebase, Heterogeneous Content and Complex Relationships

Amit P. Sheth
Wright State University - Main Campus, amit@sc.edu

David Avant

Semantic Visualization:
Interfaces for exploring and exploiting ontology, knowledgebase, heterogeneous content and complex relationships

Amit Sheth[1,2], David Avant[1]
[1]Semagix, Ltd. and [2]LSDIS Lab, The University of Georgia

Abstract

Technology platforms and products for ontology-driven process of semantic applications that serve both Enterprise wide and pan-Web needs are both available and being deployed. Similar to the development of enterprise software applications, the creation of an ontology-driven semantic application can be described as a set of distinct phases that compose a lifecycle. From conception to deployment, these phases involve human interaction with a broad variety of information, identified as heterogeneous data, metadata, knowledge and ontology. Based on experience spanning academic research at the LSDIS lab through deployed commercial semantic applications based on Semagix Freedom, this paper provides examples of graphical and visual interfaces developed to manage each phase of the lifecycle.

Introduction

Ontology-driven information systems, exemplified by research systems such as LSDIS lab's SCORE [Sheth et al 2002] are now available as commercial products such as Semagix Freedom. Research and commercial systems have now key capabilities involved in building ontology-driven information systems, which include[1]:

- development of practical ontologies with large populated domain knowledgebases: examples include TAP: http://tap.stanford.com which has rich knowledge pertaining to 12 topics, SWETO: http://lsdis.cs.uga.edu/Projects/Semdis/sweto, which is a public use ontology testbed with OWL schema, and about a million entities and over 1.5 million relationship instances
- ability to create semantic metadata in a scalable manner: examples include Semantic Enhancement Engine (now SES of Semagix Freedom) [Hammond et al 2002] for extraction from heterogeneous (structured, semi-structured and unstructured) content, and SemTag [Deill et al 2003] which is part of IBM's Web Fountain project which has demonstrated Web-scale extraction capability with extraction from over a billion Web pages
- development of semantic functional capabilities that range from search/browsing (e.g., Taalee (now Semagix) Semantic Search Engine (now SQS of Semagix Freedom) [Townley 2000], TAP Semantic Search [Guha et al 2003]), interoperability and integration (e.g., Semantic ePortals), and analytics and knowledge discovery (as in the case of a homeland security related application PISTA [Sheth et al 2004], Semagix's Anti-Money Laundering application; for more examples see [Reynolds et al 2002, and Sheth and Ramakrishnan 2003])

[1] We do not strive to provide comprehensive survey of relevant technologies and applications, but just a few we have encountered and been involved with

The development of an ontology-driven semantic application can be divided into distinct stages as illustrated below:

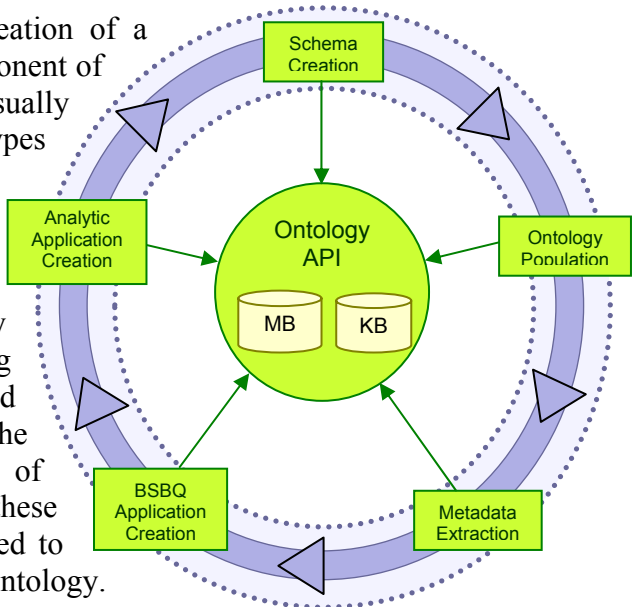


Figure 1: Semantic Application Lifecycle

The first stage of the lifecycle is the creation of a schema that serves as the definitional component of the ontology. Typical ontology schemas usually involve tens of classes and relationship types for a given application or domain (although some may be larger, depending on application scope, representation language, etc.). Examples of such applications/domains include anti-money laundering, terrorism, pharmaceutical drug discovery, Glycan structure, etc. The second task in the lifecycle is the population of the ontology at the instance level. Instances of these classes and relationships between these instances, i.e. knowledge, can be considered to be the assertional component of the ontology. In the example presented later, we use SWETO, a newly developed test bed ontology consisting of more than one million object instances and more than one million relationship instances. The next phase of the lifecycle involves the semantic annotation of heterogeneous (unstructured, semi-structured, and structured) content from a variety of sources. The process of attaching semantic annotation to a document or other piece of content is referred to as metadata extraction. Semantic applications are created by exploiting metadata and ontology with associated knowledgebase. A typical ontology-based system provides APIs to query the metadata and knowledge, and builds the application logic and GUI front end. A relatively simple example is an end-user query interface for semantic search and/or contextual browsing. One powerful, yet intuitive, interface to such a system involves a blend of semantic browsing and querying, also known as Blended Semantic Browsing and Querying (BSBQ). Using this type of interface, a user can seamlessly follow his train of thought to cross-navigate between related knowledge and content. A more advanced alternative for semantic application development could involve the creation of high-end analytical tools used for the creation of complex queries. Next, we provide small samples of various interfaces that correspond to the above; most details are skipped for brevity.

Schema Creation

The development of an ontology-driven application typically starts with the creation of an ontology schema. This schema contains the definition of the various classes, attributes, and relationships that encapsulate the business objects that model a particular domain. This model is usually devised with the help of a "domain expert" who has a deep understanding of the real-world objects and concepts in the domain. The "Knowledge Modeler" component of the Semagix Freedom toolkit (shown below) is an example of an interface that allows the user to create an ontology schema.

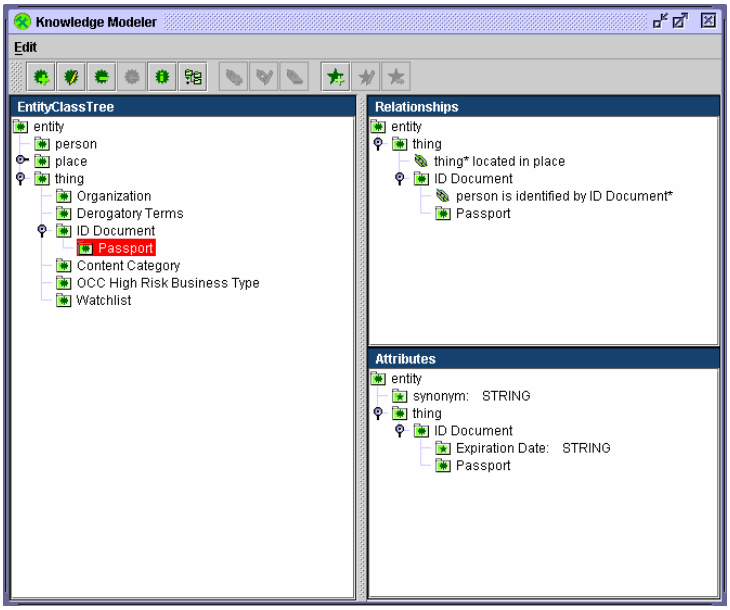


Figure 2: Knowledge Modeler

In the left pane of this tool, a user may define the ontology schema by creating a hierarchical structure of classes (similar to a directory structure). The addition of a new class as a child of an existing class indicates the "is a" relationship. After classes have been created, pairs of classes can be selected and relationships created between them (for example, "drug *has side-effect* symptom"). Properties of the new relationship such as cardinality may be specified using this interface. The available relationships for a selected class (including its inherited relationships) are shown in the top right panel of the Knowledge Modeler. The user may also select an individual relationship (for example, "person *identified by* SSN") or add an attribute definition. Attribute definitions are displayed in the bottom right portion of the interface.

In addition to the tree-based view provided by the Knowledge Modeler, the ontology schema can also be viewed as a directed graph, as shown in Figure 3.

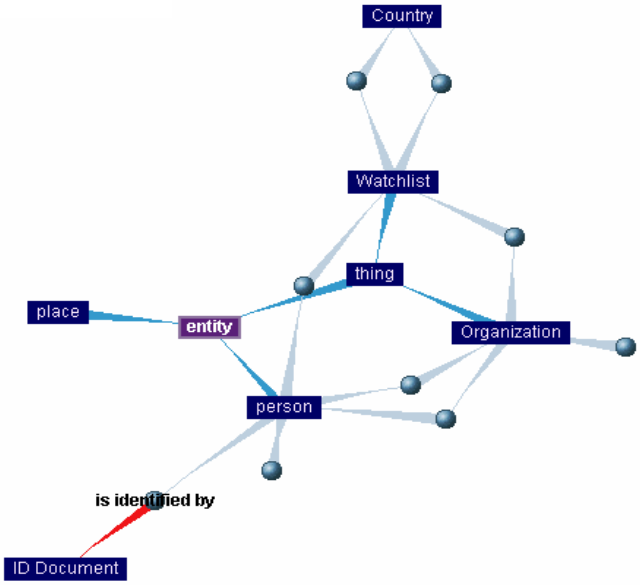


Figure 3: Graph-based ontology schema view

The schema definition for the ontology-based system not only includes the definition of classes, relationships, and attributes (the knowledge model), but also defines a set of document categories with a collection of metadata attributes for each category (the metabase model). The "Metabase Modeler" component of the Freedom toolkit provides an interface for defining this part of the system.

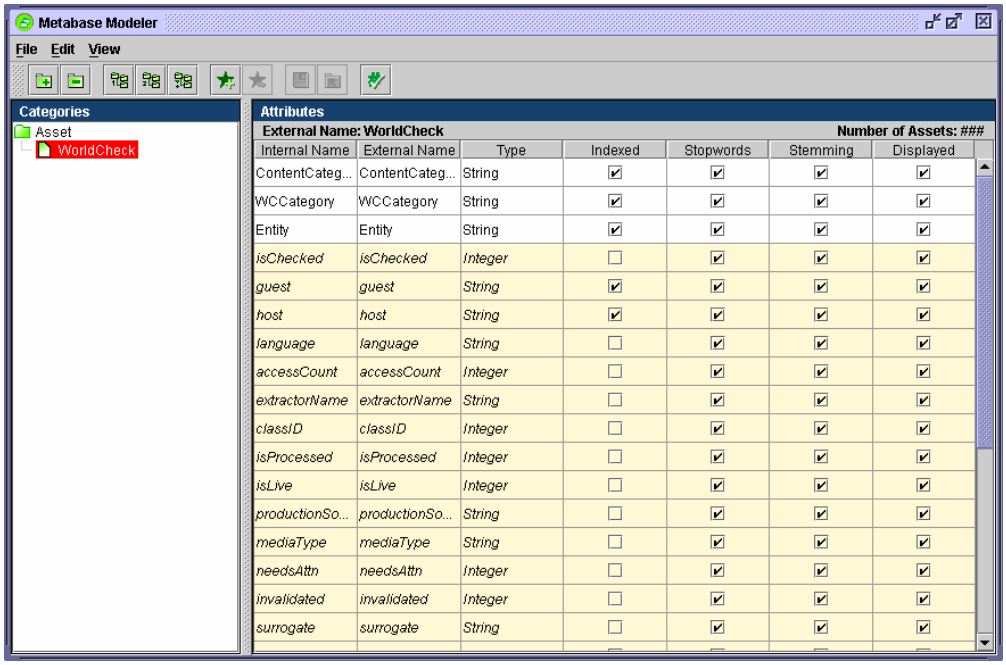


Figure 4: Metabase Modeler

Ontology Population

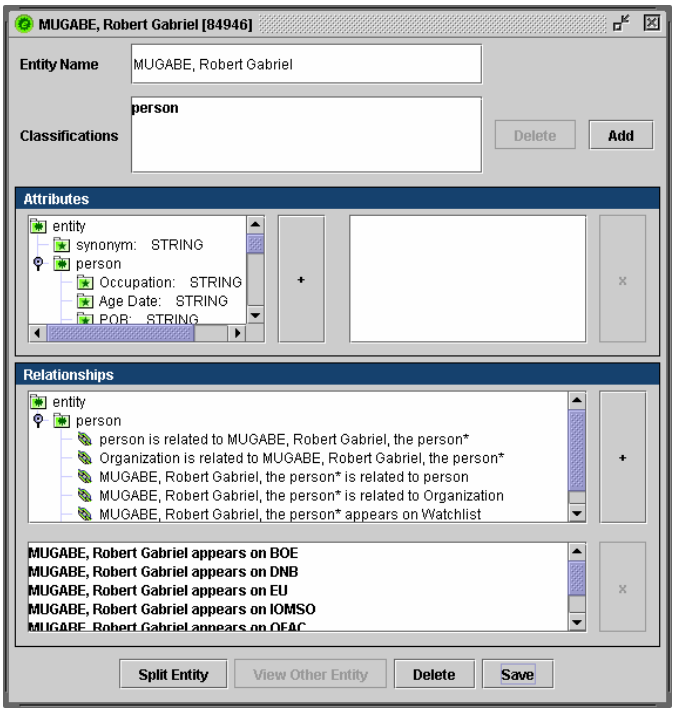
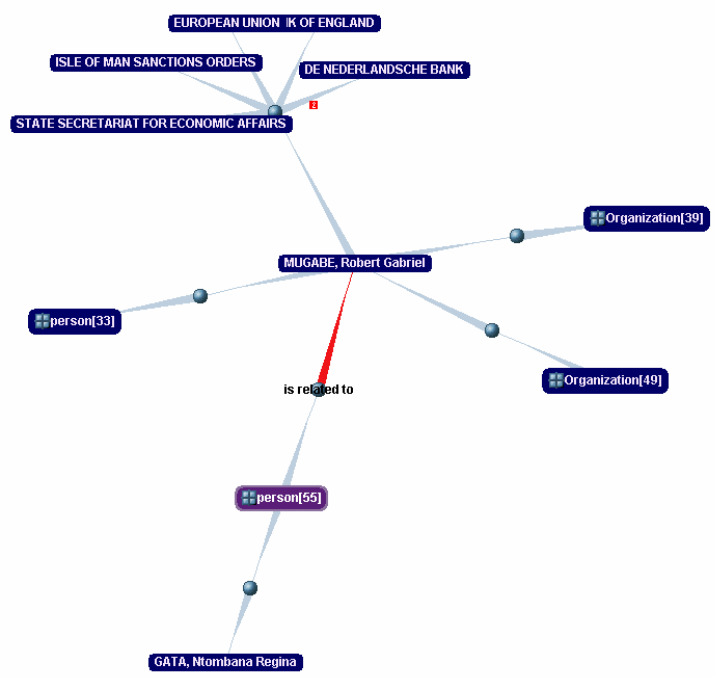


Figure 5: Entity viewer

Once the structure for the ontology has been defined, it can then be populated with instances of classes, attributes, and relationships. The collection of these instances is also referred to as a knowledgebase. In the Freedom architecture, knowledge extractor agents create the knowledgebase by retrieving information from a variety of structured and semi-structured sources. Such sources can include HTML, PDF, Microsoft Word, and XML documents as well as relational databases. After the initial knowledge extraction process is complete, the resulting knowledgebase (for a "typical" semantic business application) often contains between hundreds of thousands to a few million ontology instances. The SWETO ontology, created by the LSDIS lab at the University of Georgia, is a newly developed benchmark knowledgebase containing well over one million entities (instances of a class), relationship, and attribute instances. One of the purposes of this ontology is to expose problems that are inherent in creating real-world applications that depend on a fairly large ontology. One such problem (from a user-interface

perspective) is how to enable a user to effectively manage such a large data set. One basic, yet essential tool is an interface for viewing and editing the classification, relationships, and attributes for a single entity. The "Entity Viewer" (shown in Figure 5) component of the Freedom toolkit is an example of such an interface.

While this tool is effective for detailed information about a single entity, it is not well suited for giving the user a good overall picture of the contents of the knowledgebase. For this purpose, a graph-based view of the ontology is available.



Using this tool, the user can begin browsing the knowledgebase starting from a particular entity (the "focus" entity). Initially, only entities that are directly related to the focus entity are displayed. The user may then explore the graph in a particular direction by clicking on one of these related entities and choosing the "expand" option. In this way, the user decides what portion of the knowledgebase is relevant and obtains a better understanding of its contents by traversing the relationships between related entities.

Figure 6: Graph-based ontology instance view

Although many tools for viewing directed graphs have been created thus far, most become unusable or unintelligible when applied to real-world information. For example, it is a common occurrence to have many entities, perhaps thousands, related to a single entity via the same relationship (consider the relationship "ticker symbol *traded on* stock exchange", for example). To handle this scenario, the concept of a synthetic "collection node" was introduced. On our example, the collection node would "contain" the thousands of ticker symbol entities related to a single exchange entity. A single collection node would be related to the "NASDAQ" entity node (for example) via the "traded on" relationship. If the user wishes to see a particular member or members of the collection, those entities can be "released" from the collection by allowing the user to select these. The released entities would then be connected to the collection node with the synthetic "contains" relationship (which is not a part of the ontology schema to begin with).

Metadata Extraction

The next step in the development of an ontology-driven, semantic application often involves enhancing unstructured content (documents) with semantically relevant metadata. In Freedom, content extractor agents along with software modules called "experts" are used to perform this enhancement. The content agent first retrieves the textual contents of the document from a given source.

Figure 7: Entity and phrase detection

If the category (domain) of the document is not known a priori, it may be automatically determined using a classifier committee technique. Given the domain of the document, the expert then attempts to find entities that are explicitly mentioned in the text of the document. The following figure illustrates the detection of entities and other phrases within a piece of unstructured text. Once this set of entities is determined, a new set of inferred entities can be derived. For example, it may be inferred that a document belonging to the "business" category containing the text "MSFT" is actually about the entity "Microsoft". The expert then adds both the explicit and implicitly detected entities to a document metadata container, thus performing metadata extraction.

BSBQ Application Creation

After creating a body of semantically annotated documents (a metabase) as well as a set of inter-related ontology instances (a knowledgebase), it is now possible to create an application that will make use of both. Typical Internet users are familiar with two techniques of Web "travel" – browsing and querying. Browsing, via hyperlinks, allows users to navigate between documents that refer to each other; while searching (via Google, for example)

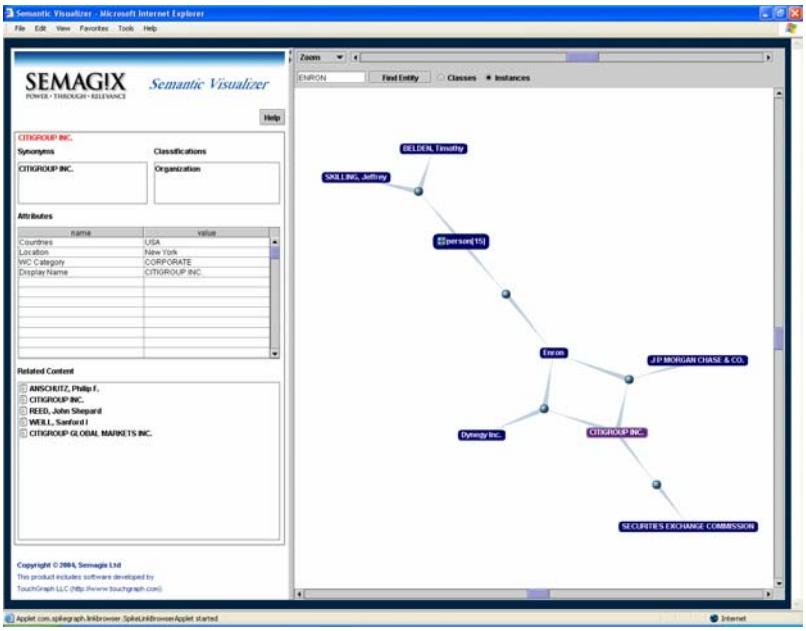
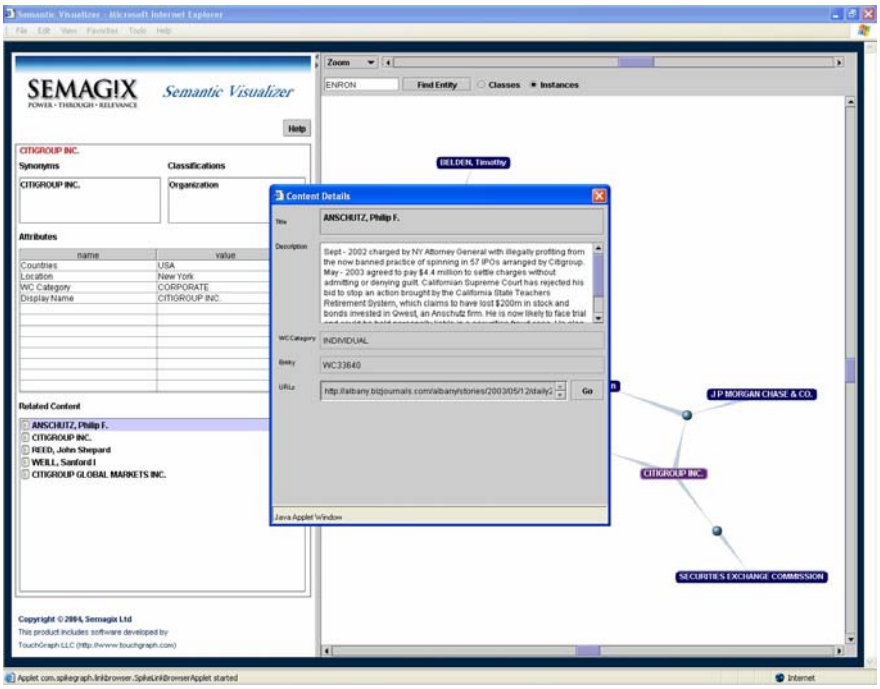


Figure 8: The Semantic Visualizer BSBQ application

"teleports" a user to an individual document. When creating a semantic application, we can combine these two techniques into one to create an intuitive, yet powerful query tool. This hybrid technique is referred to as Blended Semantic Browsing and Querying, or BSBQ. An example BSBQ application is displayed in Figure 8.

Using the toolbar at the top of the application, the user may search for the name of a specific entity in the knowledgebase. If a matching entity is found, it is displayed in the right side of the application as a directed graph. Users may view related knowledge by expanding the graph from a selected node. Each time an entity node in the graph is selected, the attribute details are displayed in a table on the top left side of the screen. In addition to attributes, all semantically relevant documents (as produced by the content extractor agents) are displayed in the list at the bottom left of the application. Double-clicking a document in the list, displays its content and relevant semantic metadata in a popup window (shown in Figure 9).



From this window, the user may select any one of the metadata items and "refocus" the graph onto that item, seamlessly following his train of thought to cross-navigate between content and knowledge.

Figure 9: Document metadata

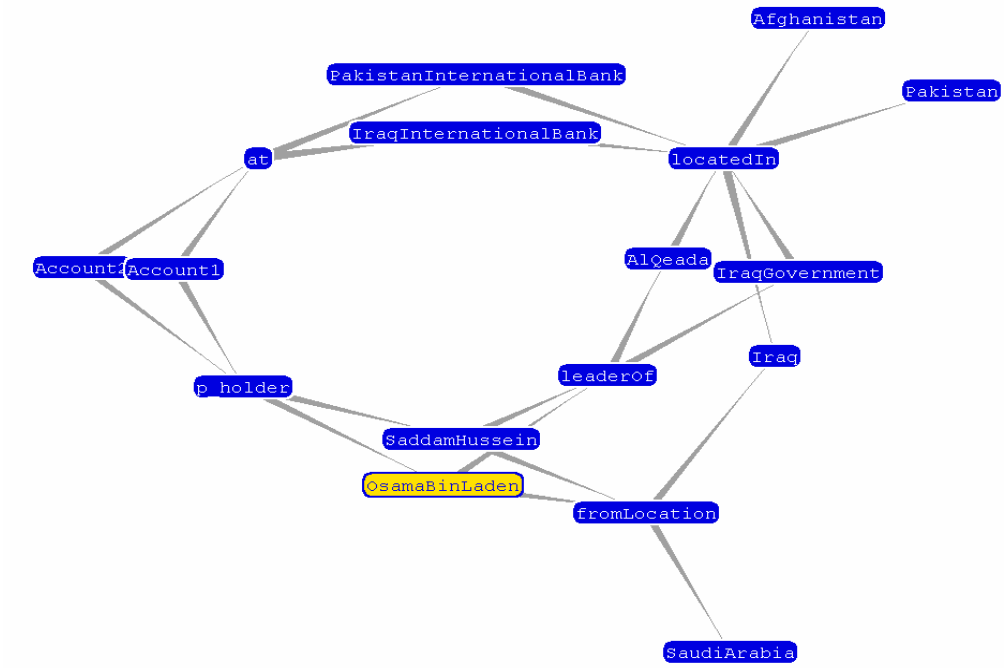
Analytical Tools

In addition to general-purpose BSBQ applications, analytical applications can be designed to provide advanced users an interface for performing ontology-specific computation as well as formulating complex queries. The CIRAS (Customer Identification and Risk Assessment Solution) Anti-Money Laundering application developed by Semagix is an example of one such application that adds an additional layer of business logic and computation to an existing semantic framework. The CIRAS analytical tools are built upon a knowledgebase containing interrelated "people" and "organization watchlist" entities possessing attributes like "address", "date of birth", etc. This application also uses a metabase of hundreds of thousands of documents containing information about individuals involved in

Figure 10: CIRAS customer risk-assessment tool.

various types of illegal activities. The CIRAS application then applies a series of rule-based heuristics to compute a "risk score" for a given individual or organization. For example, an individual who "works for" an organization that "appears on" a government-maintained watchlist would receive a higher risk score.

In a research project on Semantic Discovery[2] at the LSDIS lab and associated prototypical semantic applications for homeland security such as TRAKS [Aleman-Meza et al 2003] and PISTA [Sheth et al 2004], a TouchGraph based graphical interface has been developed to show complex relationships between entities that may meet certain patterns specified as rules or templates. Figure 11 shows some meaningful relationships between semantic metadata stored in RDF. This metadata was automatically extracted with respect to a relevant part of the SWETO[3] ontology using Semagix Freedom from distributed, heterogeneous content.



Account2 > at > IraqInternationalBank > locatedIn > Iraq Account1 > at > PakistanInternationalBank > locatedIn > Pakistan
Account2 > p_holder > SaddamHussein > fromLocation > Iraq Account1 > p_holder > OsamaBinLaden > fromLocation > SaudiArabia
Account2 > p_holder > SaddamHussein > leaderOf > IraqGovernment > locatedIn > Iraq Account1 > p_holder > OsamaBinLaden > leaderOf > AlQaeda > locatedIn > Afghanistan

Figure 11: TRAKS analytical tool.

Summary

In this paper, we provided an overview of a sample of graphical and visual interfaces used in a process employed by a comprehensive ontology-driven information system in building semantic applications. We hope this explains complex and comprehensive interplay between graphical and visual interfaces, and knowledge-based systems and processes. However, most of the interfaces are in the early phase of their evolution. Many technical challenges

[2] http://lsdis.cs.uga.edu/Projects/SemDis
[3] http://lsdis.cs.uga.edu/Projects/Semdis/sweto

remained to be addressed to improve usability, performance, scalability and functionality, which we have not discussed here for brevity.

Acknowledgement

The work presented here has benefited by contributions of many researchers at the LSDIS lab (including, B. Aleman-Meza, C. Halaschek, C. Ramakrishnan, M. Natarajan) and engineers/professionals at Semagix (including Y. Warke, C. Bertram, S. Arpinar, M. Fisher).

References

[Aleman-Meza et al 2003] B. Aleman-Meza, C. Halaschek, S. Sahoo, Terrorist Related Assessment using Knowledge Similarity, LSDIS Lab Technical Report, December 2003. http://lsdis.cs.uga.edu/proj/traks/about/fReport.html

[Deill et, al., 2003] S. Deill et. al. SemTag and SemSeeker: Bootstrapping the Semantic Web via automated semantic annotation. Proceedings of the 12th International WWW Conference (WWW 2003), Budapest, Hungary, May 2003.

[Guha et. al., 2003] R. Guha, Rob McCool and Eric Miller. Semantic Search, The Twelfth International World Wide Web Conference, Budapest Hungary, May 2003

[Hammond et. al., 2002] B. Hammond, A. Sheth, and K. Kochut. Semantic Enhancement Engine: A Modular Document Enhancement Platform for Semantic Applications over Heterogeneous Content, in Real World Semantic Web Applications, V. Kashyap and L. Shklar, Eds., IOS Press, December 2002, pp. 29-49

[IBM-WF] WebFountain, http://www-1.ibm.com/mediumbusiness/venture_development/emerging/wf.html

[Polikoff and Allemang 2003] I. Polikoff and D. Allemang, "Semantic Technology," TopQuadrant Technology Briefing v1.1, September 2003. http://www.topquadrant.com/documents/TQ03_Semantic_Technology_Briefing.PDF

[Reynold et al., 2002] Dave Reynolds , Steve Cayzer, Ian Dickinson, Paul Shabajee, SWAD-Europe: Semantic web applications - analysis and selection Appendix B - Application Survey, 2002. http://www.w3.org/2001/sw/Europe/reports/open_demonstrators/hp-applications-survey.html

[Semagix-CIRAS] Anti-Money Laundering, Semagix, Inc. http://www.semagix.com/solutions_ciras.html

[Sheth et. al., 2002] A. Sheth, C. Bertram, D. Avant, B. Hammond, K. Kochut and Y. Warke. Semantic Content Management for Enterprises and the Web, IEEE Internet Computing, July/August 2002, pp. 80-87.

[Sheth et. al., 2004] A. Sheth, et al. Semantic Association Identification and Knowledge Discovery for National Security Applications, Journal of Database Management, 2004 (to appear).

[Sheth and Ramakrishnan 2003] A. Sheth and C. Ramakrishnan, "Semantic (Web) Technology In Action: Ontology Driven Information Systems for Search, Integration and Analysis," In *IEEE Data Engineering Bulletin, Special issue on Making the Semantic Web Real, December 2003, pp. 40-48.*

[Townley 2000] J. Townley, The Streaming Search Engine That Reads Your Mind, Streaming Media World, August 10, 2000. http://smw.internet.com/gen/reviews/searchassociation/index.html.