# The Concept Object Web for Knowledge Management

James Starz, Brian Kettler, Peter Haglich, Jason Losco,
Gary Edwards, and Mark Hoffman

ISX Corporation, 4301 N. Fairfax Dr. Suite 370,
Arlington, VA 22203, USA
{jstarz, bkettler, phaglich, jlosco,
gedwards, mhoffman}@isx.com

**Abstract.** The Semantic Web is a difficult concept for typical end-users to comprehend. There is a lack of widespread understanding on how the Semantic Web could be used in day-to-day applications. While there are now practical applications that have appeared supporting back-end functions such as data integration, there is only a handful of Semantic Web applications that the average Google user would want to use on a regular basis. The Concept Object Web[1] is a prototype application for knowledge/intelligence management that aggregates data from text documents, XML files, and databases so that end-users can visually discover and learn about knowledge object (entities) without reading documents. The application addresses limitations with current knowledge/intelligence management tools giving end-users the power of the Semantic Web without the perceived burden and complexity of the Semantic Web.

## 1 Introduction

Since the creation of the Semantic Web there have been a large number of tools created that have proven its theoretical use and provided the infrastructure for application development. However, there have been very few Semantic Web applications that would be acceptable to most end users. There are many reasons this is the case as the technology is still emerging, but the foremost reason is that is currently difficult to do. Only recently have the underlying infrastructure tools matured to be used in real applications. Additionally, almost no end user will ever want to see OWL, URIs, ontologies, and the rest of the backbone of the Semantic Web.

Our experience with building and integrating Semantic Web applications has shown us the difficulties of providing functionality to users that do not care that the application uses the Semantic Web. The Concept Object Web, built on top of ISX's Semantic Object Web™ [6], is a prototype application for knowledge/intelligence management that tries to address many of the challenges of making a user friendly and useful Semantic Web application. The Semantic Object Web approach extends the Semantic Web by focusing on how users and software agents can more easily

[1] Demo available at http://semanticobjectweb.isx.com

Y. Gil et al. (Eds.): ISWC 2005, LNCS 3729 , pp. 1041 – 1049, 2005.

access and exploit information about specific entities in the world – people, places, events, etc. – that is semantically integrated from multiple distributed, heterogeneous sources. The underlying framework is used by a number of deployed applications. The Concept Object Web is based on a hybrid of features from these deployed applications using the Semantic Web.

This paper describes the basic functionality of the Concept Object Web and how it leverages the power of the Semantic Web. We discuss a number of features of the system and describe our lessons learned from implementing them. Of particular interest is how users can use the Semantic Web to manage knowledge. The system addresses issues with resolving co-references of entities across data sources. It takes into account tradeoffs for generating indices of disparate data stores for fast retrieval and inference. The paper includes Semantic Web tradeoffs on the process of gathering semantically-grounded content, indexing information, performing searches, visualizing results, discovering and browsing information, and tracking data pedigree.

## 2 Motivation and Requirements

Though there are many great tools for doing search and discovery, they often place a heavy burden on users to eventually read documents or view database records. Users have become quite willing to accept these limitations because in most cases they are not required to read more than a few documents to find the information they are looking for. Google™ works very well for the majority of users because of this assumption. In many real applications, such as business intelligence, the assumption simply does not hold. There do exist tools, such as Endeca®[2] and Siderean™[3], that do a very good job of providing typical users with the ability to perform guided searches. These may provide a better way to navigate to a smaller set of documents that would be required for an end user to read. In all of these cases, if the user wants to know all the details about an object in the document they have no choice but to read all of the documents, even if the documents themselves are mostly repetitive.

The Concept Object Web approach is to gather information from documents using automated, and when possible human, extraction of entities and facts. Additional information can come from other heterogonous sources such as databases. This information can then be represented semantically and many extraction tools are now supporting this functionality out of the box. In this setting, common URI identifiers will not automatically be given to entities, so similar object must be resolved, best they can, into a single view using existing lexical and graph matching algorithms. At this point in the process all of the information has uniform identification, as well as mappings to ontologies. This information can now be stored in a knowledge base. All of the information in the populated knowledge base can now be shown to the user for a specific knowledge object, an instance of an ontological class. Instead of reading 10 documents about a person, a user could view the aggregated information from these 10 documents in a single view. The end user can essentially read all of the

[2] http://www.endeca.com
[3] http://www.siderean.com

documents about an entity without reading all the documents[4]. The source material is retained for reference and duplicate information is rolled up. This approach has obvious advantages over other search techniques, but end users are more accustomed to reading documents and simple searching techniques. The next section describes how we bridge the gap to allow keyword search users to utilize a Semantic Web application for knowledge/intelligence management.

## 3 The Concept Object Web Application

To motivate how the Concept Object Web is relevant to knowledge/intelligence management, we present a thread of a user that highlights the functionalities of the system. This example is followed by the technical approach involved highlighting the differences between other applications and the lessons learned during development.

### 3.1 Example

Consider an analyst who is researching some suspicious activities in the Ukraine. When the analyst starts using the Concept Object Web they are presented with a sparse page with a familiar keyword search box. In this example the user makes a query for "political murders Ukraine".

The result of the keyword based search, shown in Figure 1, is a display of document metadata relevant for that query and a series of entities that are mentioned in the document collection for the query results. For the keyword query, 36 documents were returned. Along with the summary of each document, the user may view the knowledge objects, class instances, that appear in each document. The knowledge objects from the individual documents in the result set form the knowledge object display at the bottom of the screen. These entities were grouped into three customizable categories which correspond to classes in an ontology. The knowledge object portion of the display shows the occurrence count of knowledge objects that appear in the result set. In this example, the *Ukraine Parliament* occurs 6 times in our initial result set. This is somewhat interesting given our initial keyword key word query was "political murders Ukraine" and that *parliament* is one of the most frequent occurring organization in the set.

The initial query can be refined by clicking on the occurrence count for *parliament* from the previous figure. The query now consists of a keyword query and a semantic query that requires a certain entity, *parliament* in this case, to appear in the result set. The result of this refined query is a set of 6 documents that have document metadata and knowledge objects as before. The user could read those six documents, but COW provides other capabilities for exploring the information space. As was the case with the initial query, *Leonid Kuchma* is the most occurring Person in the document set. It may be of interest to find out more about *Kuchma* before proceeding. This can be done by clicking on the knowledge object link. The figure below shows the information known about *Leonid Kuchma*.

[4] The work in this paper was based on this concept introduced by Joseph Rockmore of Cyladian Technological Consulting.

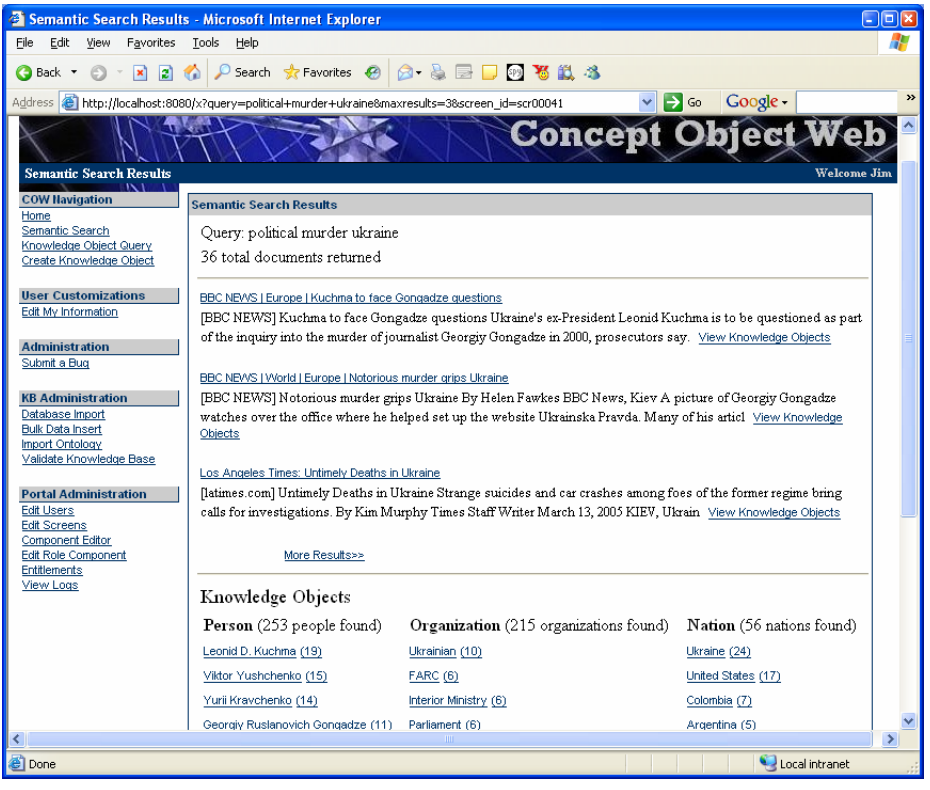


**Fig. 1.** This screen shows results for a keyword query. It includes document metadata, constrained to show three results for this picture, and knowledge objects for faceted search. The user can refine the queries using these facets or navigate to the knowledge objects themselves. The column on the left shows functions for regular users at the top and for an administrator of the system at the bottom.

This knowledge object aggregates all of the semantic information about this entity and represents it with pointers back to the source document as well as providing information about when and how each fact was obtained. Most of the assertions composing this object were created via natural language processing over the document corpus. As these assertions are just Semantic Web statements, they may come from other sources such as markup tools or relational databases.

At this point users can navigate among knowledge objects to discover new information. Additional visualizations are available supporting graph-oriented views. The application also supports a number of common Semantic Web capabilities, such as graph-based searching and pattern detection agents. For the example, the user simply navigates among related knowledge objects. The user quickly finds that *Kuchma* is accused of being involved in multiple murders. Both murders are accused by *Mykola Melnychenko* with the evidence being secret audio records. With further investigation, the accuser is a relative with *Kuchma's* rival *Victor Yanukovych* and associated with a *Russian FSB agent*. An analyst will be able to determine that *Kuchma* is being framed for these two events

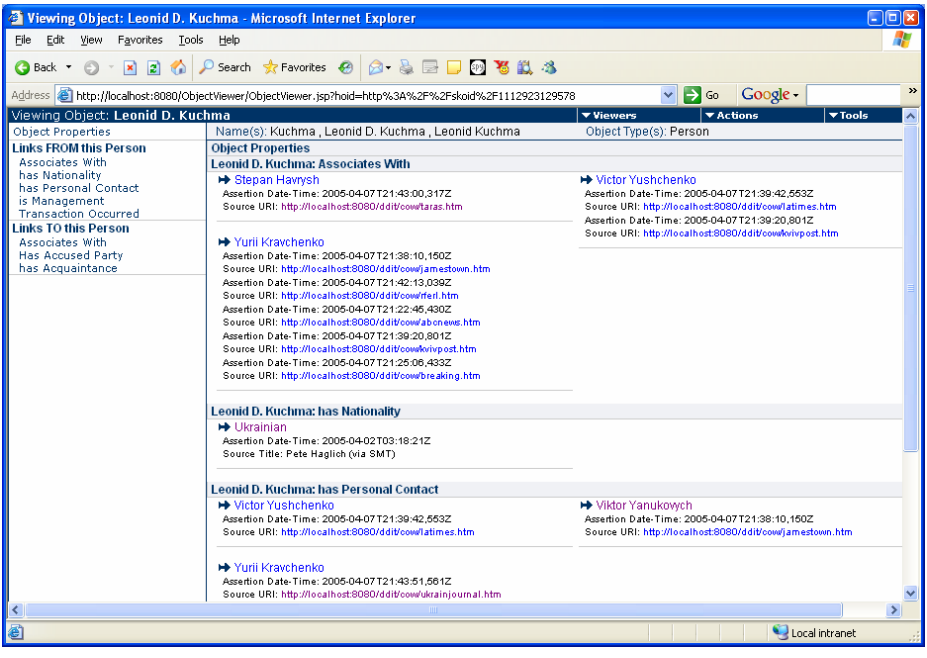


**Fig. 2.** This knowledge object displays relations and attributes for the entity. Each of the assertions displayed has metadata, which can be hidden, describing the source of the information. In this display, assertions have come from multiple sources and tools. The assertions from different sources may confirm or refute each other. The viewers tab is used for other graphical representation of the information. The actions tab allows users to edit and create knowledge objects. The tools tab is used to launch agents to look for predetermined patterns.

From our initial query concerning "political murders Ukraine" the Concept Object Web lets you refine the query with facets until you arrive at documents and knowledge objects of interest. The example discovery involves only a few hops. In the case above the human is critical in the loop to correlate information about two recordings, described textually. This could not be determined solely by a Semantic Web reasoning system. One of the key components of the Concept Object Web is to use semantics when possible, but fall back on the expertise of the user when necessary. Pointers are always available back to the source documents so users can read the original documents if they need to.

### 3.2 Technical Approach

This section describes the technical details of the system and how we dealt with the tradeoffs of Semantic Web capabilities in a knowledge/intelligent management domain.

#### 3.2.1 System Architecture

The architecture leverages tools that generate semantic markup to feed text and semantic indexing repositories. ISX's SPARKAL was used for the Semantic Web

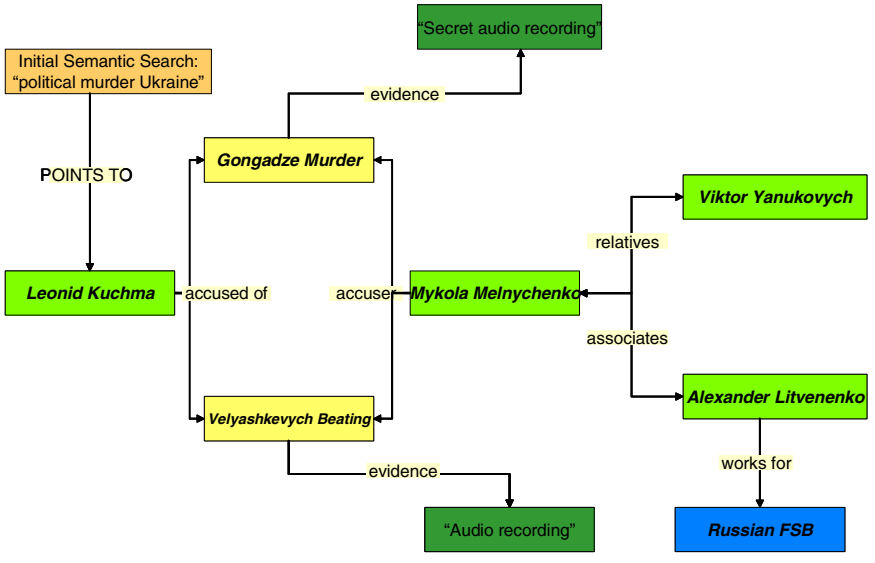


**Fig. 3.** This shows a fictitious discovery in the system concerning the interactions of Leonid Kuchma and Mykola Melnychenko. The keyword search leads to knowledge objects that help the human correlate information about the entities. In the above figure, the textual evidence is critical to the human's discovery. These types of situations cannot easily be solved by using Semantic Web technologies in isolation.

knowledge base portion and Apache's Lucene [1] was used to incorporate keyword and faceted search. As documents or databases are ingested, the Lucene index must be populated with the semantically-grounded information that results from inference in the knowledge base [4]. We chose arbitrary fields to populate for faceted search based on our ontologies, but you could conceivably take a more dynamic approach.

### 3.2.2 Search and Discovery

Over the years we have found that most users are simply not comfortable with Semantic Web style queries. This could probably be said about relational databases as well, but they are generally much more restrictive in terms of numbers of tables and fields versus ontology classes and properties. For usability, it seemed the text box was a better alternative. At that point, we could have let users search directly for knowledge objects rather than documents. We feel the use of facets, useful in their own regard, provides a conceptual jumping off point to viewing knowledge objects. The use of the facets allows the users to see how the document metadata could be leveraged.

### 3.2.3 Navigation and Visualization

Showing knowledge objects creates a number of difficulties. The most prominent of these is determining which information should be displayed to the end user. The current Concept Object Web displays all recent assertions, but other strategies are valuable. Of particular interest is aggregating data into higher abstraction levels allowing the users the ability to drill down on the information they care about while still

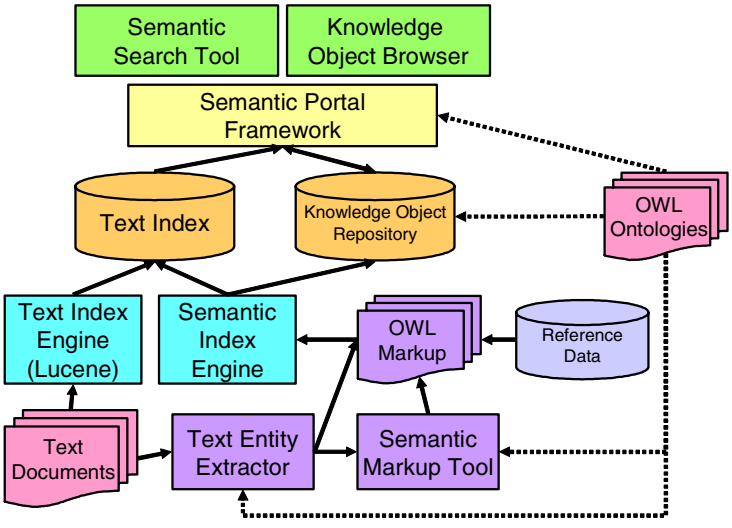


**Fig. 4.** This is the high-level architecture for COW. Text documents are indexed and processed by an entity extractor for semantic index. The portal framework provides multiple strategies for searching and discovering information.

seeing a comprehensive view of the information. There are many commercial products and research efforts that can be leveraged for visualizing graphs and networks.

The Concept Object Web primarily uses a simple web based display. Our techniques try to ensure objects have names. The first name asserted for that object will be used as the visual handle for that label. This consistency of names has been found to be extremely important. Though many objects do not have easily determined names, such as events, these unnamed objects are not terribly useful in this type of application and are usually hidden from users. To help guide navigation in the web-based view we have chosen ontological information to show in tool tip form. This has proven to help users learn information about objects within their results set without necessarily refining their query. For other navigation and visualization Inxight's StarTree™ and Graphviz are used.

### 3.2.4 Markup Generation

Markup can come from databases, XML, and text. There are increasing numbers of tools for managing relational databases as though they were semantically-grounded. Additionally, techniques are prevalent for turning XML into RDF/OWL. Text is particularly difficult to obtain markup for. Automated entity/fact extractors can provide a partial set of entities and facts, but it will miss and misclassify many entities and relations. There are a number of research-oriented tools for allowing humans to author semantic markup. These tools are good for advanced users who need to create small amounts of markup, but they can be difficult to use.

The Concept Object Web can leverage all of these types of markup despite the fact that the automated extraction may be errorful. In this application, our ontologies were developed to constrain the amount of inference that is performed. The intention is to limit the propagation of faulty data through inference. We also attempted to build our

application around classes of objects that the entity extractor is good at extracting. For the Concept Object Web, Lockheed Martin's Aeroswarm [8] was used to perform the automated extraction. To perform full exploitation of text markup, manual tools are required.

The Concept Object Web application also utilizes ISX's Semantic Markup Tool [7]. This tool leverages the use of the automatically generated markup as well as templates to aid the user. These templates act as ontological views that quickly constrain the complexity of the ontology while providing users with the ability to quickly mark up documents. Though manual creation of the markup requires resources, we have seen problems where organizations have decided the benefits were worth the cost of human effort.

#### 3.2.5 Co-reference Resolution

One of the keys to integrating information across data sources is the ability to perform co-reference resolutions across the sources. This is a very challenging problem that is not easy to address. We have found that seeding our knowledge bases with reference data containing basic information that can be used during the co-reference process is helpful.

In the Concept Object Web we seeded our system with names of popular figures for our data corpus. This data set included entity aliases and some basic information that could be used to determine identical objects. Our co-reference algorithm primarily depended on entity name matching using syntactic and phonetic cues. We used the assumption that most entities have unique names. More sophisticated graph matching could have provided better resolution performance, but for our corpus the simplifying assumptions worked sufficiently. Due to our use of automated markup generation, most entities being co-referenced had very few properties and relationships making sophisticated algorithms ineffective.

Co-reference resolution is an active area of research that cannot be covered here. Our experiences have shown us that generic algorithms for graph matching need to be supported with custom algorithms for entity types. For instance, there may be completely different algorithms for co-referencing entities and people. Syntactic similarity algorithms may work well for co-referencing people, but will lead to many false positives in co-referencing dates. We have found that each algorithm beyond name matching provides a diminishing return. Semantic Web application developers should take into account the characteristics of the data set and the required correctness for resolving duplicate entities.

## 4 Related Work

Since there are too many tools supporting knowledge/intelligence management to mention in this space, I will only refer to those that are using the Semantic Web. Most Semantic Web applications or components could easily fit into the knowledge/intelligence management framework described in the Concept Object Web. In fact, it shares many similarities to components from EU-funded research projects, such as the Information Society Technologies program, and DARPA's DAML project

[3]. It also leverages popular tools such as Jena [9] and Sesame [2] that were developed under research programs.

In terms of some popular Semantic Web knowledge management applications, there are a number of interesting applications that cover different portions of the space. SWAD-Europe [11] has produced a number of interesting Semantic Web Portal applications. They have similarities to our work, but we focus on the notion of viewing integrated knowledge objects, particularly from text data sources. Haystack [10] is comparable but leans more towards allowing clients manage their own information spaces. In our opinion tools similar to Haystack are complementary to the Concept Object Web. Semantic Search [5] does a nice job of describing integrating Semantic Web with regular search, but doesn't discuss some of the details mentioned in our work.

## 5 Conclusions

Though the Semantic Web is still emerging it is now clear that some of the barriers are being broken down to support everyday application. The Concept Object Web is an example of a new paradigm of knowledge/intelligence management tools that leverage the powers of the Semantic Web complementing the human to perform their knowledge management tasks. In this application, we believe there is added value in using the semantics to help aggregate the information and present the integrated view to users.

## References

1. Apache Lucene. http://lucene.apache.org.
2. Broekstra, J., Kampman, A., and van Harmelen, F. Sesame: A Generic Architecture for Storing and Querying RDF. Published at the International Semantic Web Conference 2002, Sardinia, Italy.
3. The DARPA Agent Markup Language, http://www.daml.org.
4. Finin, T., Mayfield J., Fink, C., Joshi, A., and Cost R. Information Retrieval and the Semantic Web. Proceedings of the 38th International Conference on System Sciences (2005).
5. Guha, R., McCool, R., Miller, E., Semantic Search. WWW2003, Budapest, Hungary.
6. Kettler, B. *et al.* The Semantic Object Web: An Object-Centric Approach to Knowledge Management and Exploitation on the Semantic Web. ISX Corporation Whitepaper. Presented as a poster at the 2nd International Semantic Web Conference (ISWC 2003). http://www.semanticobjectweb.isx.com.
7. Kettler, B., Starz, J., Miller, W., Haglich, P. A Template-based Markup Tool for Semantic Web Content. Submitted to the 4th International Semantic Web Conference (ISWC 2005).
8. Lockheed Martin 2005. Lockheed Martin AeroSWARM tool. http://ubot.lockheedmartin.com/ubot/hotdaml/aeroswarm.html.
9. McBride, B. Jena: Implementing the RDF Model and Syntax Specification. Semantic Web Workshop, WWW2001.
10. Quan, D., Huynh D., and Karger, D. Haystack: A Platform for Authoring End User Semantic Web Applications in ISWC 2003.
11. SWAD-Europe, http://www.w3.org/2001/sw/Europe/.