# Managing Knowledge for Strategic Business Analysts: The Executive Information Portal

**John Mylopoulos, Attila Barta, Raoul Jarvis, Patricia Rodriguez-Gianolli, and Shun Zhou**[1]

**Abstract.** Strategic business analysts keep track of trends that are relevant to their organization and its strategic objectives. To accomplish their mission, they monitor news stories and other reports as they become available, looking for evidence that these objectives remain on track, or have encountered obstacles. The paper presents a prototype enterprise information portal intended to support this type of knowledge work. The system supports three key functions. Firstly, it offers tools for building and analyzing semantic models of strategic objectives, the relationships among them, as well as events and actors who can play a role, positive or negative, in their fulfillment. Secondly, the system uses a powerful query language and the semantic model to assist analysts as they search external sources for relevant material, also provides semi-automatic classification and clustering of documents. Thirdly, documents are placed in an XML format and stored in an XML server. In addition, analysts can annotate or summarize documents and relate them to nodes of the semantic model.

**Keywords:** Knowledge management, semantic models, semi-structured data, information retrieval, document classification, document management.

## 1. Introduction

An important ingredient of the Information Revolution is that individuals and organizations alike have access to orders of magnitude more information than ever before. Moreover, most of this information is available in computer-based forms, including electronic documents, databases and websites. The objective of our research is to develop technologies which help a group of knowledge workers search, find, retrieve, and organize information that is useful to their daily work.

These technologies include data retrieval and analysis for structured data (i.e., database tuples and/or records), semi-structured data (such as hypertext and source code), and unstructured data (such as documents, digitized photos). In general, such technologies are bundled into a solution through an integration architecture often called an *enterprise information portal* (EIP). EIPs are "*...applications that enable companies to unlock internally and externally stored information, and provide users a single gateway to personalized information needed to make informed business decisions"* [Shilakes98]. EIPs are not off-the-shelf software. Rather, they are integrated solutions built out of components. The majority of EIP providers develop their EIP solutions from product suits offered by several cooperating partner companies. Typically, EIPs offer facilities for search, retrieval, analysis and organization of structured data and documents, along with facilities for network connectivity and computing platform interoperation.

Our approach to enterprise information portals is founded on the use of a semantic model which captures knowledge shared by a group of collaborating knowledge workers. This model is used to drive information retrieval, classification, and analysis. For example, consider a group of software engineers who own a legacy software system. An EIP solution for this group would include a semantic model which represents concepts such as (software) global architectures, call graph structures, module inter-dependencies, versions, maintenance histories, and more. Reverse engineering tools can then extract information from the legacy system and associate it with the relevant components of the semantic model. When a software engineer is working on a particular task, say fixing a bug, she can rely on the EIP (and the semantic model) to find relevant information, such as global variable declarations, who calls the procedure that is being debugged and more. This is precisely the approach explored in [Finnigan97].

We are currently developing a prototype toolset, called the Executive Information Portal (or EXIP), intended to help a group of strategic business analysts working for a large corporation. Their task is to keep track of current events as they unfold, and make sure that their company's strategic objectives remain on track. To work on this task, business analysts scan news stories (Reuter's, CNN, etc.), analysts' reports (Yankee Group, Forrester Research, and the like) and other document sources, looking for relevant material. Once they have decided that a particular document is useful, they add it to their own library, write annotations and prepare memos to be circulated to their colleagues. This work is currently done without any tool support, or vanilla computer tools (e.g., a web browser and search engine). Our toolset aims to support the semi-automatic search and classification of documents, also the analysis of collected information with respect to a given set of strategic objectives.

[1] Authors' address: Department of Computer Science, University of Toronto, Toronto, CANADA; {jm,atibarta,prg,jarvis,szhou}@cs.utoronto.ca.

The development of our toolset was guided by a number of quality requirements. Firstly, we wanted a system which evolves semi-automatically in the sense that its model and classification scheme may be modified by users, or automatically, thanks to the application of Machine Learning techniques. In addition, the classification of downloaded information is also assumed to evolve, along with the model and the classification scheme. Finally, we expect that our system will be used with minimal feedback from users, who are busy people and can't afford to spend much time training the system's classifiers and other functions.

Section 2 of the paper introduces the architecture of the proposed system and motivates its features, while section 3 describes the tools used to build wrappers for external information sources. Sections 4 and 5 present the concepts used to build and visualize semantic models, also their use for modeling and analyzing strategic objectives. The document management subsystem is outlined in section 6, while section 7 presents the retrieval, classification and clustering facilities supported by the prototype. Conclusions and directions for further research are presented in section 8.

## 2. System Architecture

The EXIP global architecture is shown in figure 1. The outermost layer of the architecture (bottom part of the figure) includes external information sources, such as CNNfn (cnnfn.cnn.com/news/technology/), CBC business news (cbc.ca/business/), the Globe and Mail (globeandmail.com/hubs/rob.html) and Forrester Research, whose reports we assume that the analysts download manually. This layer also includes wrappers for each source, which specify expected outputs for input queries. The information sources may have formats that are structured or semi-structured, proprietary or public domain (e.g., plain text.) In the prototype implementation, we wrap semi-structured (HTML) sources and plain text sources only. Structured sources (such as relational data) are easy to wrap and access, while documents in proprietary formats (PDF, MS-Word, RTF, etc.) can be translated to a common HTML or XML format using commercial tools such as Document Navigator from Verity [Verity].

It is important to note that documents transformed to HTML from a proprietary format are different from "native" HTML documents in that the former contain only useful information, while the latter contain an abundance of noise (advertising, site navigational menus, etc.) It is the job of the wrapper to locate the information of interest in a page for a given query, and separate it from such noise.

Another significant difference between the two types of documents is that those in a proprietary format are very likely to be local, easy to access files. In contrast, "native" HTML documents are often buried deep in a website and require considerable navigation in order to be retrieved. To specify the navigational path that will lead to a document to be retrieved, we use WebOQL, a second-generation web query language [Arocena98].

Wrappers export data in XML format. One important reason for selecting XML over HTML is that XML is fast becoming the lingua franca for information exchange, and its adoption allows us to benefit from a wealth of research in this area. Moreover, XML tags carry more useful information than HTML ones. This information can be used to better index and otherwise process retrieved documents. All exported data are translated into a common document schema, essential for the processing of documents after they have been downloaded.

The Document Clustering and Summarization component attempts to classify all downloaded documents by relating them to one or more elements of the semantic model. The proposed classification may be approved or overruled by the users of the system, or it may be accepted "as is" when the system is in "automatic" mode. The Document Management component, DocMan, provides support for document viewing, annotation and manipulation. DocMan uses an XML Data Server for its operations.

ModViz is the Model Visualization component. It offers support for creating, visualizing and maintaining the semantic model. Using ModViz, strategic analysts can view different parts of the semantic model, update it, or retrieve all relevant documents associated with some of its elements.

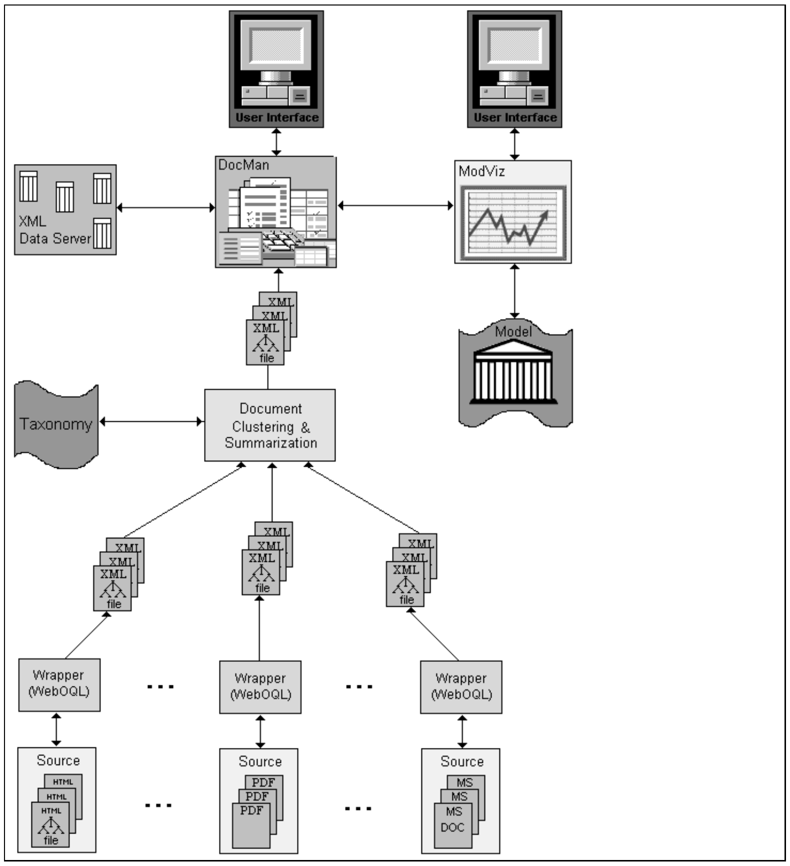


**Figure 1**: The global architecture of the EXIP.

## 3. Wrapping External Information Sources

News data is an essential information source for strategic business analysts. Given that all major news providers have a Web presence, we need tools that facilitate the creation of wrappers for news sites. For our prototype, these tools are founded on WebOQL, a second generation Web query language. WebOQL is actually used both for path navigation and for filtering.

Most of the HTML pages in large sites are generated from templates. Because of the complexity of these templates, web programmers often use comments to facilitate page maintenance. Figure 2 shows a WebOQL query which takes advantage of comment information. The query says something like "SELECT w.url FROM target WHERE w is child of y, y is child of x and x is <TR> anywhere in the document and y is <TD> and w is <A> and y has a <comment> child that contains "Newspaper Content begins". Thus, this query calls for a search of a table row anywhere in the document that contains the comment 'Newspaper Content begins', followed by a navigational path where tag <TD> is nested in a row tag <TR> and so on. This query extracts the links to the news files related with the "Report on Business" section from the Globe and Mail website.

For documents that do not contain useful comments, we have to adopt a different approach. Suppose, for example, that the HTML page from the example above does not contain the comment "Newspaper Content begins". In this case, we use WebOQL's path navigation capability to traverse the structure of the document that is being queried. In support of this capability, WebOQL offers path regular expressions along with a set of operators for searching trees.

```
target := browse("http://www.globeandmail.com/hubs/rob.html");

SELECT [w.url]
   FROM x in target via ^*[tag = "TR"], y in x', z in y', w in y'
   WHERE y.tag = "TD" and w.tag = "a" and  z.tag = "comment"
                 and z.source ~ "Newspaper Content begins"
```

**Figure 2**: A WebOQL query.

Evaluation of a WebOQL query proceeds by first building an Abstract Syntax Tree (AST) for the page that is being queried, followed by tree traversal operations, as specified by the query. Figure 3 shows an example of the AST for an HTML page, while figure 4 shows an example WebOQL query.

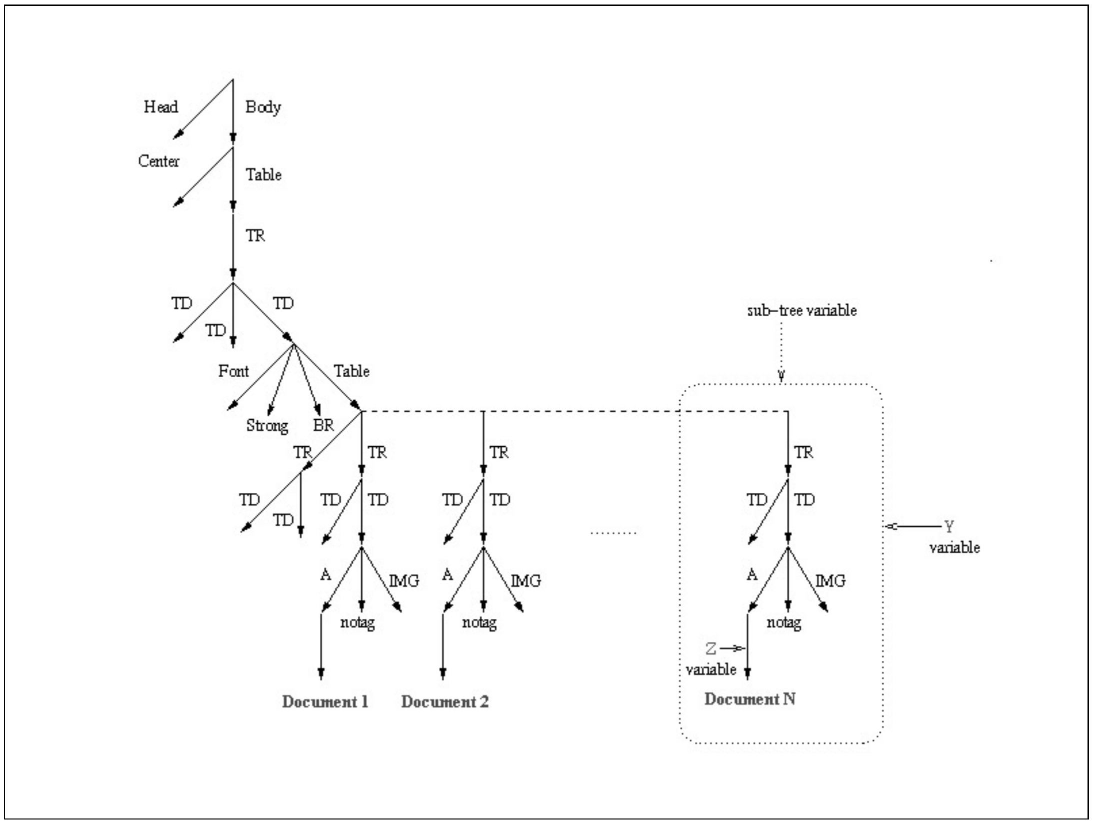


Figure 3: An AST for an HTML page.

The query says something like "starting from the root of the tree, find a 'table' tag; then move down 3 levels (apply the subtree operator three times) and assign the nodes you are at to y; then, move down 2 levels and take the first child (use of the head operator)." The operators used in the query are:

"'" (apostrophe) -- the subtree operator , returns a subtree for its operant node;
"&" (ampersand) -- the head operator , returns the first child node for its operand;
"!" (exclamation mark) -- the tail operator, returns an enumeration of all but the first children nodes of its operand.

```
target := browse("http://www.theglobeandmail.com/hubs/rob.html”);

SELECT [z.url]
    FROM x IN target  VIA ^[tag = "table"],
               y IN (((x'')!!)')!!!)', z in (((y')!)')&;
```

**Figure 4:** Navigational query in WebOQL.

## 4. The Semantic Model

The semantic model provides a description of the strategic objectives of an organization in terms of goals and subgoals, also the events and actors that can influence any of these goals (positively or negatively). The model can be thought as a network of relationships between goals, actors, events and documents. Thanks to the rich modeling framework, the model supports various forms of analysis. For example, the analyst can visualize if and how the organization is advancing towards achieving its goals, what are the obstacles, critical events to look out for, and what are the dependencies to external organizations.

A goal represents a desirable state of affairs, such as “Be the largest internet service provider (ISP) in Canada”, or “increase market share by 20%”. A goal can be decomposed into subgoals through two basic types of relationships. An AND-relationship relates a goal to a set of subgoals such that fulfilling all subgoals is a sufficient condition for the fulfillment of the goal. OR-relationship relates a goal to a set of subgoals such that fulfilling at least one of the subgoals is a sufficient condition for the fulfillment of the goal. These are well-known concepts in AI planning and problem solving. For strategic planning purposes, however, it is often the case that lateral influences between two goals. For instance, the

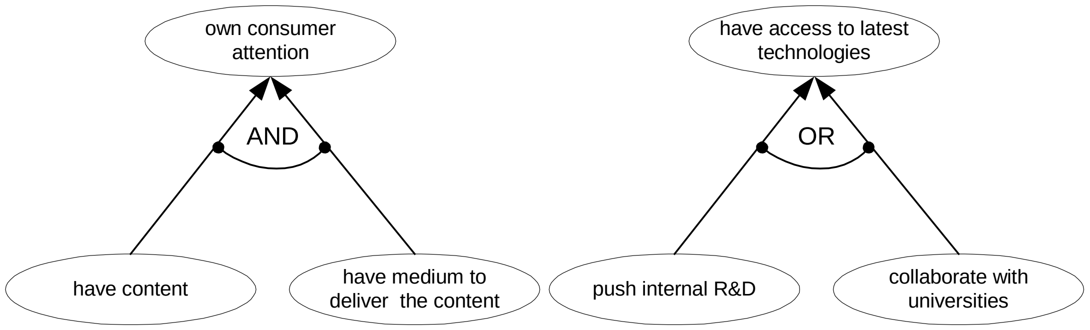

own consumer attention
AND
have content
have medium to deliver the content
have access to latest technologies
OR
push internal R&D
collaborate with universities

**Figure 5:** AND- and OR-decompositions of goals.

goal “push internal R&D” may influence negatively the goal “reduce costs” which have been generated as one of the subgoals of the goal “increase profits”. Such influences are represented in terms of effect relationships, which can be labelled “+” or “-“, depending on the nature of the influence (figure 6).

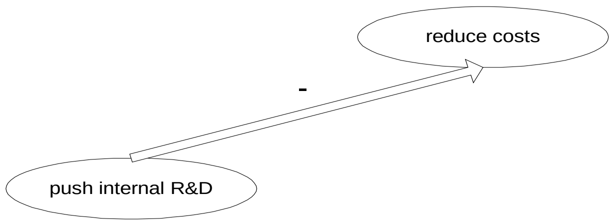


**Figure 6:** A negative lateral effect relationship among goals.

A business analyst can use goal decomposition to reduce an organizations strategic objectives down to smaller and better defined and medium- to short-term tactical subgoals. Alternative solutions for satisfying a goal can be explored and described through OR-decompositions. Decompositions of top-level goals also helps to identify possible conflicts or synergies between their subgoals. A more elaborate example of goal decomposition is shown in figure 7.

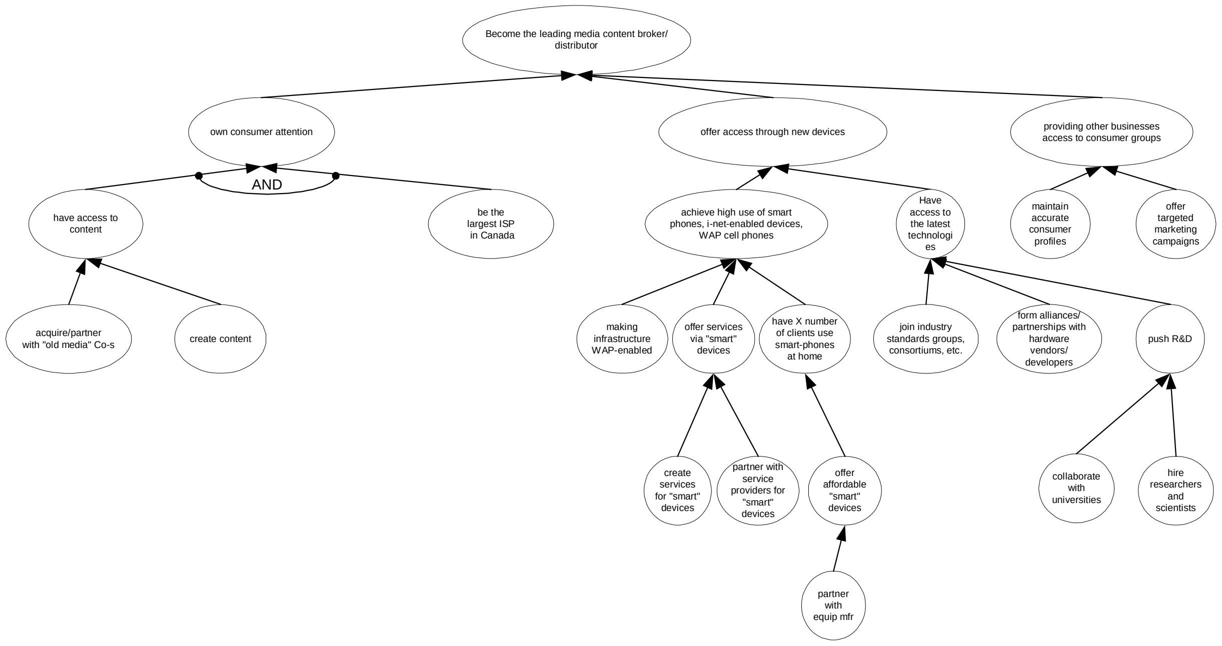

Become the leading media content broker/ distributor
own consumer attention
offer access through new devices
providing other businesses access to consumer groups
AND
have access to content
be the largest ISP in Canada
achieve high use of smart phones, i-net-enabled devices, WAP cell phones
Have access to the latest technologies
maintain accurate consumer profiles
offer targeted marketing campaigns
acquire/partner with "old media" Co-s
create content
making infrastructure WAP-enabled
offer services via "smart" devices
have X number of clients use smart-phones at home
join industry standards groups, consortiums, etc.
form alliances/ partnerships with hardware vendors/ developers
push R&D
create services for "smart" devices
partner with service providers for "smart" devices
offer affordable "smart" devices
collaborate with universities
hire researchers and scientists
partner with equip mfr

**Figure 7:** Goal decomposition example.

Goals are related to events in several different ways. Events may cause the fulfillment of a goal, or contribute to the fulfillment of a goal positively or negatively. Events may also serve as road signs, indicating how far from fulfillment is a particular goal. Business analysts base their analysis of strategic objectives on reported events. The semantic model describes what types of events are relevant to each goal and how they affect it. This description can then be used to determine what events to look for in external information sources.

For our purposes, an event is an occurrence of an activity. For example, “Telecom A buys media company B for $1B” is an example of an event. Events are related to goals through positive (“+”) or negative (“-“) links. In the semantic model, we want to describe not just events as they occur, and their impact on goals, but also generic events, such as “acquire a competitor”, which describe what sort of information to look for.

Apart from being related to goals, events can also have causal or temporal relationships with other events. These relationships indicate that one event can cause another, or one relationship precedes/follows another. For example, the event “financial difficulties for media company” causes “media company stock plunges”, or “sign a deal with equipment manufacturer” temporally precedes “expand operations”. Unlabelled links among events represent “causes” relationships, while “precedes” links represent temporal relationships.

Events can also strengthen/weaken a relationship between two other artifacts (goals, events). For example, Figure 8 shows an event “competitor starts to offer internet service” which strengthens a subgoal relationship between two goals.

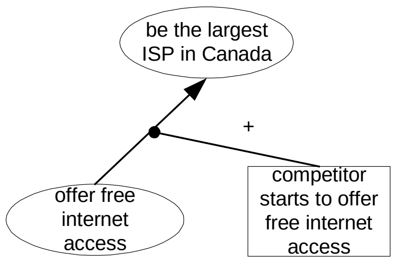

be the largest ISP in Canada
+
offer free internet access
competitor starts to offer free internet access

**Figure 8:** Positive influence of a change event towards a subgoal relationship.

Apart from goals and events, the modeling framework also includes the concept of an actor. An actor is an organizational unit, such as a business partner, supplier, internal division, or department. Actors can be associated with goals through a dependency relationship (satisfying a goal depends on an actor) or through an interest relationship

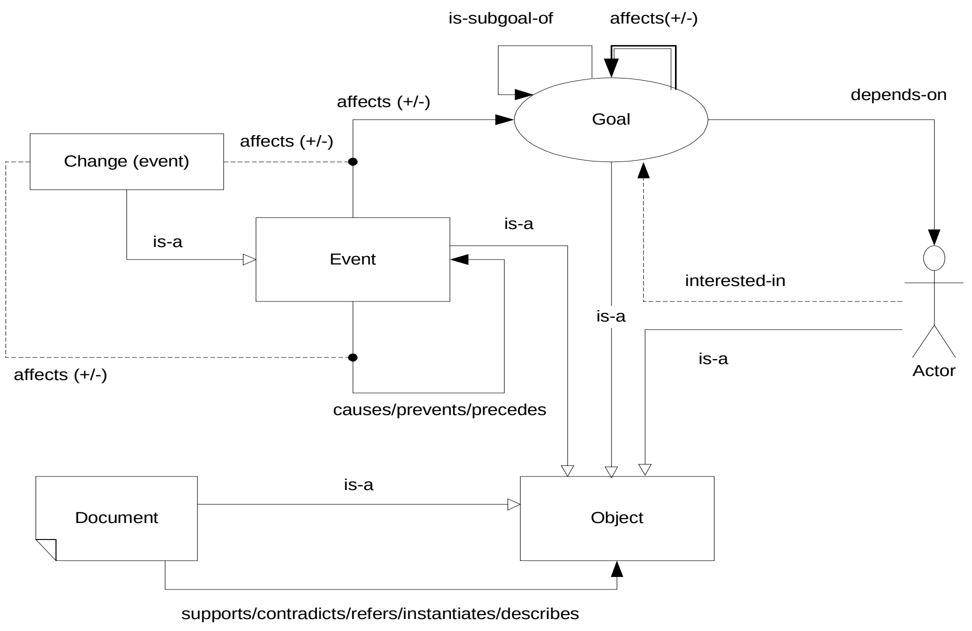


**Figure 9:** The EXIP metamodel.

(which indicates that satisfaction of a goal is of particular concern to an actor).

A summary of the modeling concepts supported by the EXIP framework is shown on figure 9. Finally, figure 10 shows a complex strategic model for a hypothetical telecom company's strategic objective "become a leading media broker/distributor".

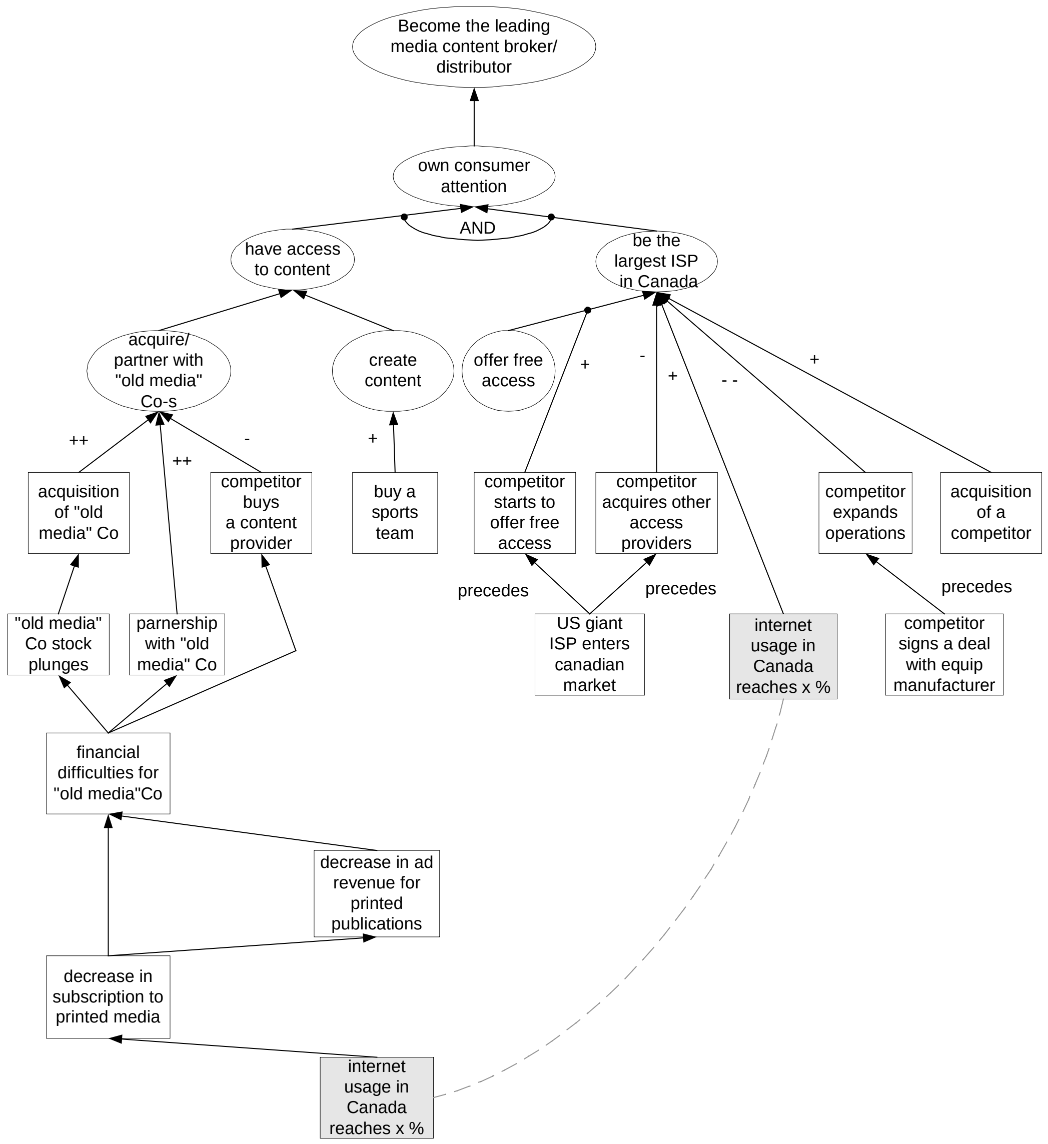

Become the leading media content broker/ distributor
own consumer attention
AND
have access to content
be the largest ISP in Canada
acquire/ partner with "old media" Co-s
create content
offer free access
++
++
-
+
+
-
+
- -
+
acquisition of "old media" Co
competitor buys a content provider
buy a sports team
competitor starts to offer free access
competitor acquires other access providers
competitor expands operations
acquisition of a competitor
precedes
precedes
precedes
"old media" Co stock plunges
parnership with "old media" Co
US giant ISP enters canadian market
internet usage in Canada reaches x %
competitor signs a deal with equip manufacturer
financial difficulties for "old media"Co
decrease in ad revenue for printed publications
decrease in subscription to printed media
internet usage in Canada reaches x %

**Figure 10:** Sample semantic model for the goal "become a leading media broker/distributor".

Our modeling framework is an adaptation of the *i** modeling framework [Yu93, Yu95, Yu96] intended for modeling social settings. The goal analysis framework is based on goal analysis techniques presented at [Mylopoulos01]. A general survey of modeling techniques is offered in [Mylopoulos98].

## 5. Visualizing the Semantic Model

The shared knowledge of a group of analysts, represented by a semantic model, can be quite large and complex. Moreover, users of the EXIP (that is, business analysts and executives) are usually concerned with strategic planning from different perspectives. To aid the understanding and acquisition of knowledge at different levels of granularity, the user interface of EXIP supports a variety of visualization techniques for browsing/exploring the model as well as the contents of the EXIP. In this section, we outline these techniques.

Since the model can potentially contain hundreds of nodes and edges, users need facilities to filter away parts of the graph. The graph viewer offers a number of *views* of the semantic model. Each of these has a unique id and name, along with a criteria description, that permit its easy identification. The criteria description reflects the type of the view (built-in vs. user-defined view) and the attribute-based specification used to create the view. The following list outlines some of these views:

- *Global view*: displays a graph view of the complete model; this unrestricted view reveals the complexity of the knowledge captured by the model.
- *Goal interaction view*: shows a higher-level graph view of the model with only the top strategic goals and any lateral dependencies among them. This view offers executives a simplified but concrete perspective of the company's current strategic plan.
- *Actor contribution view*: presents a graph view highlighting the contribution dependencies (i.e. "interested-in" and "depends-on" links) among actors and goals in the semantic model. This view permits to weight the role played by external or internal actors in the company's current strategic plan.
- *Top goal view*: displays a graph view for each top strategic goal in the system. Each view represents a sub-graph of the global view (any artifacts, any relationships) starting at a given top goal. These views isolate main strategic objectives from the global plan and facilitate goal analysis and monitoring done by business analysts. Figure 11 shows the resulting top goal view for the goal "Become the leading media content broker/distributor".

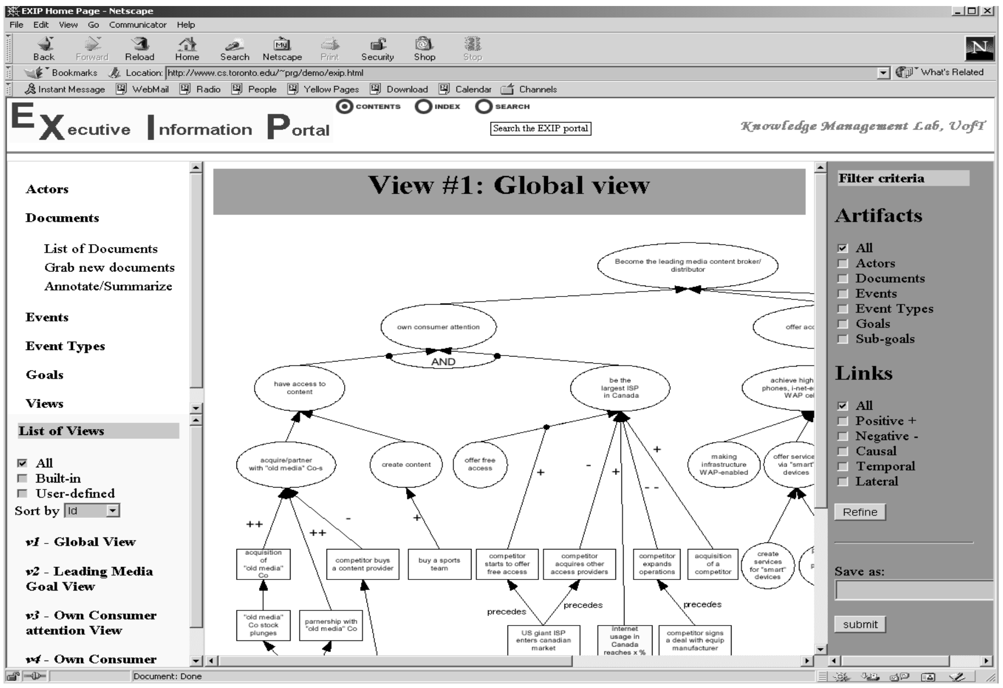

EXIP Home Page - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://www.cs.toronto.edu/~prg/demo/exip.html
What's Related
Instant Message WebMail Radio People Yellow Pages Download Calendar Channels
EXecutive Information Portal
CONTENTS INDEX SEARCH
Search the EXIP portal
Knowledge Management Lab, UofT
Actors
Documents
List of Documents
Grab new documents
Annotate/Summarize
Events
Event Types
Goals
Views
List of Views
All
Built-in
User-defined
Sort by Id
v1 - Global View
v2 - Leading Media Goal View
v3 - Own Consumer attention View
v4 - Own Consumer
View #1: Global view
Become the leading media content broker/distributor
own consumer attention
AND
have access to content
be the largest ISP in Canada
achieve high
acquire/partner with "old media" Co-s
create content
offer free access
making infrastructure WAP-enabled
acquisition of "old media" Co
competitor buys a content provider
buy a sports team
competitor starts to offer free access
competitor acquires other access providers
competitor expands operations
acquisition of a competitor
create services for "smart" devices
"old media" Co stock plunges
parnership with "old media" Co
US giant ISP enters canadian market
Internet usage in Canada reaches x %
competitor signs a deal with equip manufacturer
precedes
++
-
+
- -
Filter criteria
Artifacts
All
Actors
Documents
Events
Event Types
Goals
Sub-goals
Links
All
Positive +
Negative -
Causal
Temporal
Lateral
Refine
Save as:
submit
Document: Done

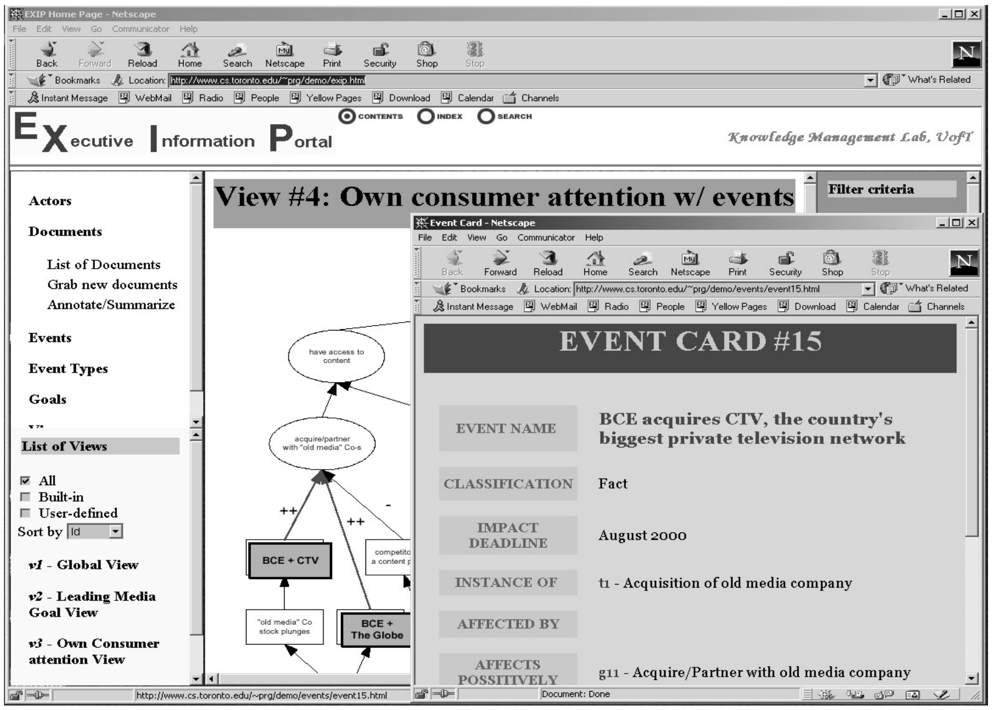
**Figure 11**: Top goal view for goal "Become the leading media content broker/distributor"

- *Event type view*: displays a graph view for each event type in the model. Each view shows a sub-graph of the global view centered on a given event type node and with all its immediate neighbors (i.e. all goals affected by this template, all its instances and any other event types linked to it through temporal or causal dependencies). This view summarizes the contribution of event types to the strategic plan; business analysts may find it useful to know which event types have not yet been instantiated, so that they can focus their search to these event types.

In addition to the built-in views provided by the system, the graph viewer supports *layout manipulation*, *hypertext*, *exploration* and *filtering* capabilities on the displayed portion of the graph. These operations permit users not only to accommodate the displayed view according to their mental view of the system but also to further explore the details of the semantic model.

*Layout* operations provide users with the ability to define different graph layouts for the displayed portion of the model. Using different graph layout algorithms (such as spring, Sugiyama-style and hierarchical) present users with other kind of visual information that could prove helpful. For example, a spring or Sugiyama layout can provide some evidence to business analysts that a number of selected nodes in a global graph view form a cluster of highly interconnected nodes (goals and events) in the strategic plan.

Graph *exploration* facilities allow exploring the current view of the semantic model by increasing or decreasing levels of details (both laterally and vertically) at a current node. After selecting a current node in the view (for example, an event instance node), users can choose different actions from a pull down menu. The set of actions available at a given spot depends on the type of the current designated node. For example, if the chosen node is an event instance node, then the user can choose to either show/hide its sub-tree (other events or supporting documents) or show additional details about it (i.e. event card description). Figure 12 shows the result of choosing the second action for the "BCE + CTV" event node in the displayed view.



EVENT CARD #15

| | |
|---|---|
| EVENT NAME | BCE acquires CTV, the country's biggest private television network |
| CLASSIFICATION | Fact |
| IMPACT DEADLINE | August 2000 |
| INSTANCE OF | t1 - Acquisition of old media company |
| AFFECTED BY | |
| AFFECTS POSSITIVELY | g11 - Acquire/Partner with old media company |

**Figure 12**: "Own consumer attention with events" view and event card for event "BCE + CTV"

To dynamically reduce the number of visible elements in a graph view, users can apply *filtering* operations. The EXIP graph viewer offers two types of filters: *node* and *edge type filters*. The first one lets users choose what kind of artifacts they are interested in (for example, only goal or sub-goal nodes); the second one allows focusing on particular relationships of the already chosen types of nodes (for example, only those dependencies that impact positively other goals). Figure 13 depicts the result of applying a node type filter to the top goal view shown in Figure 11.

As indicated earlier, filtering operations allow users to simply filter unnecessary details and alter focus. They do not affect the underlying semantic model. However, users can save filtered results into re-usable views. *User-defined views* allow business analysts not only to save their current point of interest in their daily work but also to promote some kind of collaborative working among other users. A view is physically stored by only saving the *filter criteria* used to create it. This means that any modification to the model will potentially impact the number of artifacts and links in each saved view (both built-in and user-defined).

## 6. Document Management

DocMan, the document management component of the EXIP, is designed to manage thousands of documents that have multiple links among them. Furthermore, we have to support complex search operations on these documents, involving both full-text and metadata search. Moreover, the documents have to be indexed according to a multidimensional index with respect to the semantic model. Last but not least, the system has to support instant updates.

A possible platform for the implementation of DocMan is provided by information retrieval system technologies. Such technologies can handle large collections of documents (up to a range of millions). In addition, these technologies are also equipped with advance search capabilities such as fuzzy or proximity search. However, we decided that such technologies are too "heavy weight" for our purposes. Moreover, they don't support real-time updates of documents, nor do they use in a direct way hypertext links.

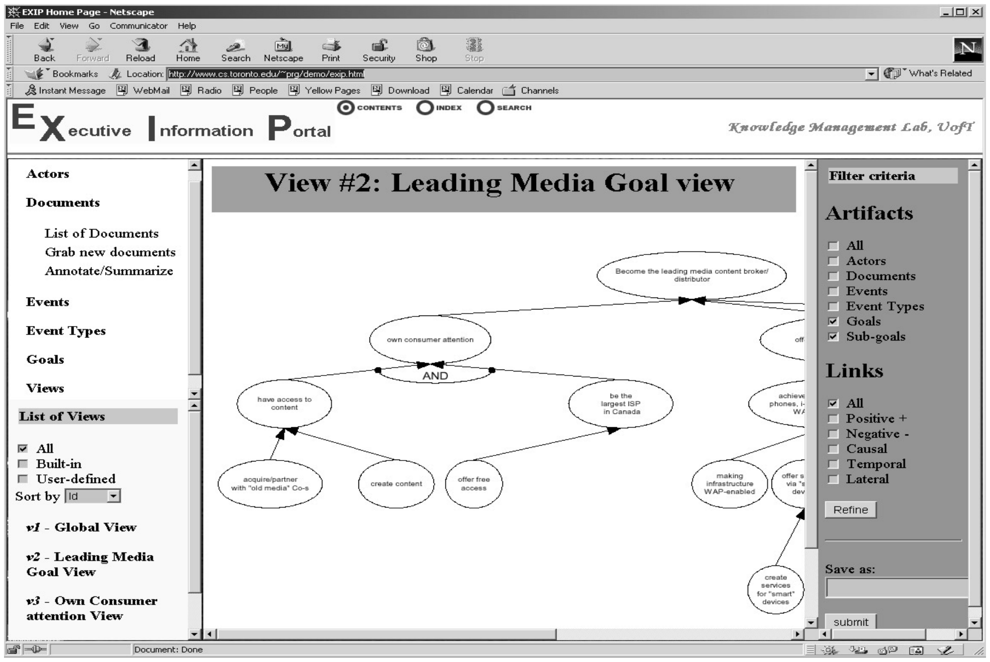


**Figure 13:** "Leading Media Goal" view with filtering

At the other end of the spectrum there are document content management systems, also sometimes known as document management systems. Most of these were developed for technical publishing management, but recently they moved into the enterprise portal domain. With respect to our requirements, such systems do support hyperlinks between documents; also they have facilities for describing document metadata. Furthermore, the latest versions of such systems have moved towards the adoption of XML as their document model, meeting another one of our requirements. Their main drawback, however, is the fine granularity at which they work. We would like to manage documents at the document level, rather than the paragraph or images level. Furthermore, DCMSs do not address issues of efficient storage and retrieval for documents.

Because of the aforementioned requirements, we propose to adopt ToX (the Toronto XML Server) as the backbone of our document management system. ToX is currently being developed at the University of Toronto. The principal objective of the ToX project is to offer database-like services and performance to XML data/documents processing, including data catalogues, alternative storage methods, complex query processing, full-text and path index capabilities, as well as transaction processing. Ongoing research efforts concentrate on querying XML data sources, and physical storage for XML data. Commercial products offer diverse solutions for managing XML data but these are generally rather primitive and inflexible. Thus, object-relational DBMS vendors use their products' richer data types to accommodate XML data, while object-oriented DBMS vendors simply accommodate XML data as another complex object. Neither of these two groups offer a solution specifically designed for XML data and processing requirements.

One of these particular requirements lies in the "stucturedness" of XML documents. Thus, we would like distinct data representation and processing for relatively unstructured documents (such as a Shakespearean play), versus highly structured (…and less interesting…) documents originating from a relational database. Distinct treatment is desirable not only with respect to physical storage alternatives, but also with respect to indexing and querying. For existing systems, the user can specify distinct treatments for different types of documents through scripting. For example, IBM's DB2 XML-Extender [DXX], the user can specify through the Document Access Definition (DAD) file a particular storage and indexing alternative. In this process, the user has no input to the "stucturedness" of the document, thus her chosen storage alternative is subjective and may not be close to an optimal solution. Furthermore, the user has to have prior knowledge of the complex DAD syntax. In contrast, the ToX project proposes to offer relational DBMS like capabilities whereby the user registers XML data and the system provides a set of primitives that assures optimal storage and indexing.

Storage for the XML documents is still an open research issue. One of the obvious alternatives is to store an XML document in several relational tables according to same mapping strategy. Different mapping alternatives have been proposed. Just to mention a few, in [Florescu99] the authors propose the edge approach, the binary approach and the universal table. In the edge approach, each edge is represented as a tuple in a table. The tuple records the source and target node id as well as the edge label, while the element and attribute values are stored in a distinct table. In the binary approach, a separate table is created for each distinct element and attribute that appear in the document and store the element/attribute values in this tables. In the universal table approach, all edges/values are stored in one relational table. In the Monet XML project [Schmidt00] the authors propose storing each distinct path in a separate relation. A similar approach is used by Dataguides [Goldman97]. Unlike the other proposals, Dataguides are used mainly as an index structure rather than a storage structure.

Choosing a query language for the system is much easier than choosing a storage alternative. The research community proposed a wealth of XML query languages (see [Bonifati00] and [Simeon00] for surveys). Our final choice was between XML-QL [Fernandez99] and Quilt [Carey00]. Although the two languages have the same expressive power, we picked Quilt because it is more "readable" and it has a well-known implementation in Kweelt [Sahuguet00]. We could also use WebOQL as the XML query language. Arguably, WebOQL has the same expressive power as XML-QL and it is already used in a different part of the EXIP. However, WebOQL was design with the Web in mind. Although, it can query any type of semi-structured data (including XML), its main function has been querying the Web. On the other hand, XML query languages proposed so far do not handle well inter-document references (links in the HTML case). Many of the XML query languages do not support links directly, though they do support pointers (IDREFs). Links could be implemented on the top of these pointers. Therefore, our decision was to use what is better suited for the job, WebOQL for wrapper building and Quilt for querying XML data.

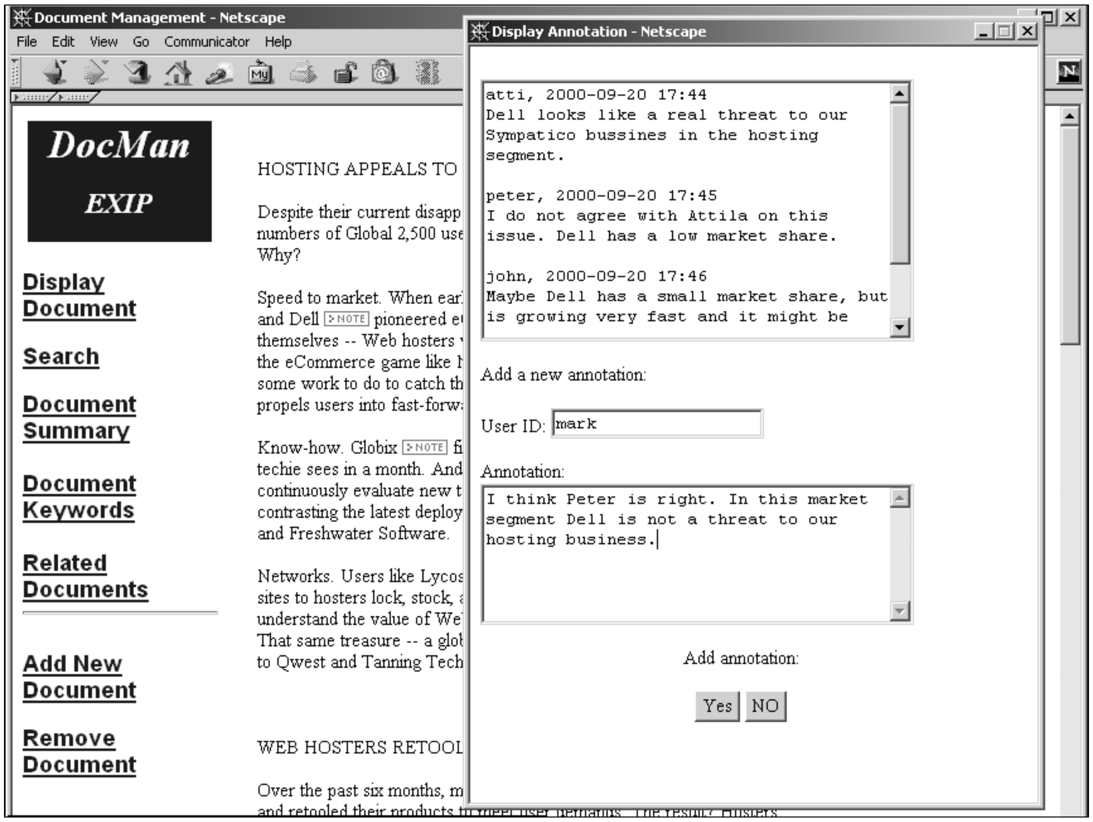
Given that ToX is in the early stage of development, we have adopted for the first DocMan implementation off-the-shelf software for our XML server storage component. In particular, we use IBM's DB2 XML-Extender [DXX]. In addition we plan to test and evaluate other off-the-shelf XML server solutions, such as the Poet content management system [Poet] and the Microsoft SQL-Server 2000 [MS-SQL].

The basic functionality of DocMan includes registering and storing documents, as well as support for document annotation and document summary. In addition, DocMan supports primitives for document-related EXIP functions such as relationships across documents and document search (full text, metadata, and index based search.)

Once a document is registered in DocMan, it can be annotated or summarized. Annotations can be thought as "knowledge records" for our system in the sense that they facilitate the exchange of knowledge between a group of collaborating business analysts. Moreover, annotations can be used as discussion threads. Figure 14 shows such a discussion thread. Furthermore, DocMan allows annotations to be broadcast, via e-mail, to the entire analyst group. Document summaries constitute another important form of knowledge sharing. A document might have many summaries attached which represent the perspectives of many analysts who have reviewed the document. In addition, DocMan offers support for attaching keywords to registered documents. This function can be used in order to update the list of keywords attached to any one document by the classification component of EXIP.

Annotations, as well as summaries and keywords are all encoded in XML. Annotations and summaries are separate XML documents, linked to the document they refer. The attached keywords are incorporated in the original document, now in XML format. Parts of the annotation and summary documents are the username of the analyst that authored the annotation/summary. We need this information in order to search and mine the data with respect to the author.

Another DocMan feature is its support of semantic relationships among documents. For example, a document may support or contradict or follow-up or simply relate to another document. Such relationships are inserted by users and are part of the EXIP semantic model. In the internal XML representation of documents, a relationship consists of a relationship tag with the corresponding attribute for the relationship type.



HOSTING APPEALS TO

Despite their current disapp
numbers of Global 2,500 use
Why?

Speed to market. When ear
and Dell NOTE pioneered e
themselves -- Web hosters
the eCommerce game like N
some work to do to catch th
propels users into fast-forw

Know-how. Globix NOTE fi
techie sees in a month. And
continuously evaluate new t
contrasting the latest deploy
and Freshwater Software.

Networks. Users like Lycos
sites to hosters lock, stock,
understand the value of We
That same treasure -- a glob
to Qwest and Tanning Tech

WEB HOSTERS RETOOL

Over the past six months, m

```
atti, 2000-09-20 17:44
Dell looks like a real threat to our
Sympatico bussines in the hosting
segment.

peter, 2000-09-20 17:45
I do not agree with Attila on this
issue. Dell has a low market share.

john, 2000-09-20 17:46
Maybe Dell has a small market share, but
is growing very fast and it might be
```

User ID: mark

Annotation:

```
I think Peter is right. In this market
segment Dell is not a threat to our
hosting business.
```

**Figure 14:** Annotations in DocMan.

Since annotations and summaries are represented in XML format, they are easy to search. Thus, we use the ToX search engine to search not only documents but also annotations and summaries. Moreover, using the XML structure we can also follow the relationships among documents.

## 7. Retrieval, Classification and Clustering

As indicated earlier, analysts search for information relevant to the objectives of their organization in pre-selected document sources, such as news services (Reuter's, CNN, etc.), analysts services (e.g., the Yankee Group or Forrester Research) and more. Once they have concluded that a particular document is relevant to their work, they download it, summarize it and classify it within their workspace. This section demonstrates how the EXIP improves this document retrieval and classification procedure by adopting Machine Learning techniques for information retrieval, classification and clustering. The section first postulates a number of requirements for the retrieval, classification and clustering component of the EXIP. We then present technical details of our classification algorithms, also demonstrate their effectiveness with preliminary experimental results.

The main requirements for this component of the EXIP is to reduce manual labor by providing semi-automatic retrieval, classification and clustering of documents. For retrieval, we expect that analysts will identify document sources, and someone with technical background will generate wrappers, using the techniques described in section 3. For classification, analysts only define the strategic business model and provide a small number of sample documents, from which the system generates a set of classifiers which classify automatically documents that are downloaded through different wrappers. Documents that are not relevant to any element of the semantic model are filtered out. For clustering purposes, analysts select a threshold. Once the number of documents associated to a node in the model is greater than the threshold, the system applies a clustering algorithm which splits the document set into subsets. Thus, clustering helps analysts to evolve the strategic business model. Analysts can also give their own extension of the model to refine the classification scheme, instead of relying on automatic clustering.

```
<!ELEMENT EXIPDOCUMENT (TOPICS, MODEL, DOCUMENT) >
<!ELEMENT TOPICS (PCDATA) >
<!ELEMENT MODEL (GOALS, EVENTS, LINKS) >
      <!ELEMENT GOALS (#PCDATA) >
```

```
    <!ELEMENT EVENTS (#PCDATA) >
    <!ELEMENT LINKS (#PCDATA) >
<!ELEMENT DOCUMENT (TITLE, SOURCE, DATE, AUTHOR, CONTENT, RELEVANTDOCS) >
    <!ELEMENT DOCID (#PCDATA) >
    <!ELEMENT TITLE (#PCDATA) >
    <!ELEMENT SOURCE EMPTY >
      <!ATTLIST SOURCE NAME CDATA #REQUIRED
                URL CDATA #REQUIRED >
    <!ELEMENT DATE (#PCDATA) >
    <!ELEMENT AUTHOR  (#PCDATA) >
    <!ELEMENT CONTENT  (#PCDATA) >
    <!ELEMENT RELEVANTDOCS (RELEVANTDOC+) >
    <!ELEMENT RELEVANTDOC EMPTY >
      <!ATTLIST RELEVANTDOC DOCUMENTID CDATA #REQUIRED
                  LINKTYPE CDATA #REQUIRED >
```

**Figure 15:** The Document XML schema

Document flow within the EXIP is summarized in figure 1. Documents are transformed from a domestic format used within each document source into an XML schema. The schema that has been adopted is given in figure 15. The schema consists of three sections. The first is the TOPICS section, used for general topic classification of the document. The second is the MODEL section which is used for model classification. Here the GOALS, EVENTS, and LINKS subsections assign a document to nodes in the semantic model. The third is the DOCUMENT section, and it includes relevant document fields, such as TITLE, AUTHOR and the like.

Once wrappers have selected documents from a particular document source, the documents are parsed and transformed into the EXIP XML format. The transformation is necessary because documents from different sources may have very different formats, such as Acrobat PDF, HTML, or pure ascii. Even if they are in the same format, say HTML, they could still present information in very different ways.

Classification is based on supervised Machine Learning techniques which need training data, i.e., documents which have been classified and labelled by experts. In selecting a classification technique, we are assuming that strategic business analysts don’t have much time to train classifiers. Accordingly, we need a classifier which will perform reasonably after being trained with a small number of sample documents.

For classification purposes, the semantic model is treated as three taxonomies for goals, events, and links. Each document can be associated with zero or more nodes or links within each taxonomy. For example, a news document from the the Toronto Star is first classified within the goal taxonomy, then the events taxonomy, and finally the links taxonomy.

There already exists a large -- and increasing -- number of classifiers, such as nearest neighbor (kNN), inductive rule learning, linear least squares fit (LLSF), neural networks, support vector machines (SVM), decision trees, naïve Bayesian (NB) classifiers and more. Among them, SVM, kNN, and LLSF have been suggested as top performers. For our purposes, we have chosen kNN as our classifier because it performs well even with small training sets. Moreover, its algorithm is quite simple, compared with the competitors, and is therefore easily modifiable so that it works as a continuously trainable classifier. kNN is an instance-based learning algorithm which has been studied in Pattern Recognition for a long time [Dasarathy91]. In the document classification community, it is also known as memory-based reasoning (MBR) [Masand92]. The kNN classifier is generally considered as one of the top-performers among classifiers [Iwayama95, Yang99]. During our experiments, we also experimented with NB, because it is computationally economical, and performs surprisingly well in many application areas [McCallum98]. Unfortunately, the results we got when using NB were substantially worse than those for kNN, so kNN was adopted as our classifier of choice.

A kNN classifier assumes that documents can be represented as vectors representing points in an n-dimensional Euclidean space. Each dimension represents one keyword and the *i*-th coordinate of a document vector represents the number of times keyword *i* appears in the document, divided by the total number of keywords in the document. The distance between document vectors $\alpha$ and $\beta$ is defined to be

$$d(\alpha, \beta) = \sqrt{\sum_{i=1}^{n} (\alpha(x_i) - \beta(x_i))^2}$$

or simply the cosine value for the two vectors

$$\cos(\alpha,\beta) = \frac{|<\alpha,\beta>|}{|\alpha\|\beta|}$$

Training documents are labelled with predefined categories. Given a query document, the classifier finds out the k nearest neighbors and takes a straw poll on these classifications. The categories which have higher values than a predefined threshold are suggested target categories for the new document.

A classifier works on a text corpus which is divided into a training set and a testing set. The classifier has an adjustable classification scheme which is trained with documents in the training set. The trained classifier is then used to classify documents in the testing set. It is usually assumed that classifiers are not trained during the testing phase. This is a limitation we would like to remove. For the EXIP, we'd like to start with a small training set, but continue the training during the testing and operation of the classifier. Moreover, we'd like to have reasonable performance at the end of the (short!) training phase. To meet these requirements, we had to adopt the standard kNN classifier.

kNN classifiers postpone most computation until a query document comes in and has to be classified. The training of a kNN classifier basically involves storing training documents, which are represented as document vectors. When a testing document is being classified, the classifier searches through the whole training set to find the k nearest neighbors. The target categories are suggested by a vote of these neighbors.

In order to make a kNN classifier trainable not only during initial training but also during its lifetime, we need to make two adjustments. First, the classifier should be able to add documents into its training set even after the initial training phase. That is, when an analyst finds a misclassified document, she can correct the classification and give as feedback to the classifier the correct categories for the document. In addition,, we need to impose an upper limit to the training set size, otherwise the training set will keep growing, slowing down the whole system. Once the training set reaches the limit, a replacing policy is applied when the classifier is presented with additional feedback.

As new documents are downloaded and classified with respect to the model, the number of documents associated with some nodes might become too large, making such document groups unmanageable. To solve this problem, we offer clustering techniques which split big document groups automatically into subgroups of similar documents. Our clustering algorithm is based on the group average clustering algorithm introduced in [Cutting92].

Let G be a document group. The group average similarity of G is

$$S(G) = \frac{1}{|G|(|G|-1)} \sum_{\alpha\in G}\sum_{\alpha\neq\beta} s(\alpha,\beta)\cdot$$

Let Φ be a set of disjoint document groups. The algorithm searches for two different groups Γ and Δ which maximize S(Γ∪Δ) over all choices from Φ. A new partition Φ' is constructed by merging Γ with Δ

$$\Phi' = (\Phi - \{\Gamma',\Delta'\}) \cup \{\Gamma \cup \Delta\}$$

Initially, Φ is a set of singleton groups, one for each individual document to be clustered. The iteration terminates when |Φ'| equals the desired number of clusters. The complexity of the clustering algorithm is $O(n^2)$ where n is the number of documents to be clustered. Since our system is expected to handle a maximum of a few thousand documents, such complexity is acceptable.

The following graphs show the performance of our kNN classifier, with the Reuters-21578 corpus as the benchmark, in which only documents that belong to the top ten categories have been used. There are a number of performance measures for classifiers, such as precision, recall, F1, and break even point. Break even point has been used in our experiments, which is the value where precision and recall equal to each other.

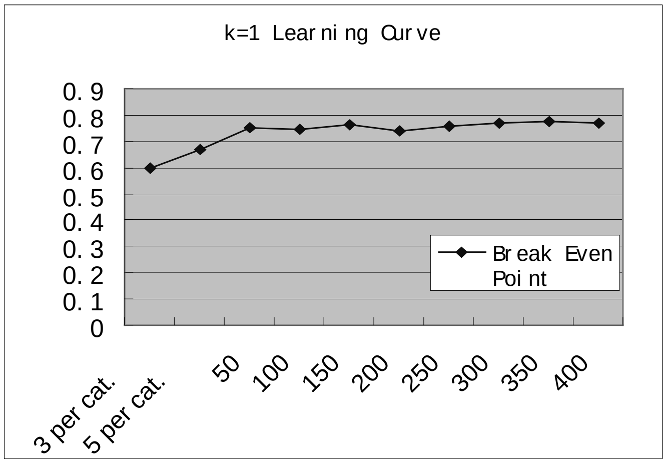


Figure 16: kNN Learning Curve (k=1)

We trained the classifier with two initial training sets which have 3 and 5 documents for each category respectively, then we tested the classifier on a testing set with 2000 documents. The figure shows the classifier achieves break even points, 0.6 and 0.67 from 3-document-per-category and 5-document-per-category initial training set respectively. Compared with other published results, these results are low, but also interesting because the training set size of other experiments is oncludes generally hundreds or even thousands of documents per category.

After trained with the training set with 5 documents per category, we allow the classifier to accept feedback of its classification results, i.e. the classifier is told what are the correct categories if it classifies a document incorrectly. Misclassified documents are added into the initial training set until the number of training documents reaches an upper size limit. We impose different upper limits to the traning set size, from 50 to 400. Once the classifier reaches the upper limit, it is tested on 2000-document testing set. The results in Figure 16 show that classifier performance improves quickly with taking a small number of misclassified documents. For example, we have break even point 0.753 of the training set size 50 and 0.764 of the training set size 150, an increase of 0.083 and 0.094 over that of 5-document-per-category. However, when we keep adding more misclassified documents, performance gets satuated gradually. In other words, relatively small feedback at the beginning of a classifier's operation pays off. But beyond a certain point, such feedback does little to further improve performance.

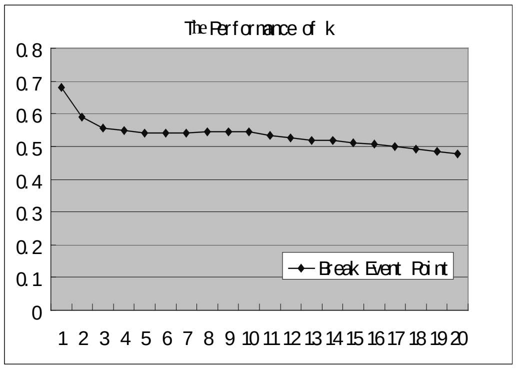


Figure 17: Performance of the kNN classifier as a function of k

Which k performs the best? Theoretically, a small k is expected to perform better in our system considering the small size of the training set. Figure 17 confirms the prediction. The classifier in the figure is trained by a initial training set of 5 documents per category and performs a test on a 2000-document testing set. The figure shows clearly that k=1 performs the best and performance decreases as k increases.

## 8. Conclusions

We have presented a prototype EIP intended to support a group of knowledge workers in retrieving, classifying and using information downloaded from a variety of information sources. Our approach differs from state-of-practice EIPs in that it treats all information managed by the system as semi-structured data, thereby exploiting tools such as declarative query languages and XML data servers. Moreover, our approach adopts a Machine Learning framework for quick learning and continuous evolution of its classifiers and clustering algorithms. Finally, our framework supports lightweight semantic models of relevant domain knowledge which is used for classification, retrieval and analysis.

We are currently completing the implementation and evaluation of the prototype. In addition, we are working on the definition of goal analysis techniques, such as evaluating the satisfiability of top-level goals, given a set of event instances about the application domain. Another form of analysis under review involves the identification of critical goals and events. These are goals or events which have become key to the fulfillment of top-level goals and about which additional information is needed.

**Acknowledgements**

The research reported in this paper has been funded by the Bell University Laboratories (BUL), the Natural Sciences and Engineering Research Council (NSERC) of Canada, and the University of Toronto. We are grateful to Geoff Riggs

(Bell Canada) for introducing us to the world of strategic business analysis, also to Michael Milton and Adele Newton (BUL) for helpful direction and moral support.

## References

[Arocena98] Arocena, G., Mendelzon, A., “WebOQL: Restructuring Documents, Databases and Webs”, Proceedings ”, Inernational Conference on Data Engineering (ICDE’98), Florida, 1998.

[Barta98] Barta, A., *WebCat Information Integration for Web Catalogs*, Master's Thesis, Dept. of Computer Science, University of Toronto, 1998.

[Bonifati00] Bonifati, A., Ceri, S., “Comparative Analysis of Five XML Query Languages”, *SIGMOD RECORD, 29(1),* January 2000.

[Carey00] Carey, M., Florescu, D., Ives, Z., Lu, Y., Shanmugasundaram, J., “XPERANTO Publishing Object-Relational Data as XML”, Proceedings of the Third International Workshop on the Web and Databases, 2000.

[Chamberlin00] Chamberlin, D., Robie, J., Florescu, D., “Quit: An XML Query Language for Heterogenous Data Sources”, Proceedings of the Third International Workshop on the Web and Databases, 2000.

[Cutting92] Cutting, D.R., Karger, D.R., Pedersen, J.O., Tukey, J.W., “Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections,” Proceedings Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 318 – 329, 1992.

[Dasarathy91] Dasarathy, B.V., “Nearest Neighbor (NN) Norms : NN Pattern Classification Techniques,” McGraw-Hill Computer Science Series, IEEE Computer Society Press, Las Alamitos California, 1991.

[Deutsch99] Deutsch, A., Fernandez, M., Suciu, D., “Storing Semistructured Data with Stored”, Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data, 1999.

[DXX] http://www-4.ibm.com/software/data/db2/extenders/xmlext/

[Domingos97] Domingos, P., Pazzani, M., “On the Optimality of the Simple Bayesian Classifier Under Zero-One Loss,” *Machine Learning (29)*, 103-130, 1997.

[Fernandez99] Deutsch, A., Fernandez, M., Florescu, D., Levy, A., Suciu, D., “A Query Language for XML”, Proceedings of the Eight International World Wide Web Conference, 1999.

[Finnigan97] P. Finnigan, I. Kalas, K. Kontogiannis, H. Müller, J. Mylopoulos, S. Perelgut, M. Stanley, K. Wong, “The Software Bookshelf”, *IBM Systems Journal,* November 1997.

[Florescu99] Florescu, D., Kossmann, D., “Storing and Querying XML data using RDBMS”, IEEE Data Engineering Bulletin, 22(3), 1999.

[Friedman97] Friedman, N., Geiger, D., Goldszmidt, M., “Bayesian Network Classifiers,” *Machine Learning (29),* 131-163, 1997.

[Goldman97] Goldman, R., Widom, J., “Dataguides: Enabling Query Formulation and Optimization in Semistructured Databases”, Proceedings of the 23th International Conference on Very Large Databases, 1997.

[Iwayama95] Iwayama, M., Tokunaga, T., “Cluster-Based Text Categorization: A Comparison of Category Search Strategies,” ACM International SIGIR Conference on Research and Development in Information Retrieval, 273-281, 1995.

[Manolescu00] Manolescu, I., Florescu, D., Kossmann, D., Xhumari, F., Olteanu, D., “Agora: Living with XML and Relational”, Proceedings of the 26th International Conference on Very Large Databases, 2000.

[Masand92] Masand, B., Linoff, G., and Waltz, D., “*Classifying News Stories Using Memory Based Reasoning*,” In Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 59-64, 1992.

[McCallum98] McCallum, A., and Nigam, K., “A Comparison of Event Models for Naive Bayes Text Classification,” AAAI Workshop on Learning for Text Categorization, 1998.

[MS-SQL] http://www.microsoft.com/catalog/display.asp?subid=22&site=10145.

[Mylopoulos98] Mylopoulos, J., “Information Modeling in the Time of the Revolution”, *Information Systems 23(3-4),* June 1998, 127-156.

[Mylopoulos01] Mylopoulos, J., Chung, L., Liao, S. S. Y., Wang, H., Yu, E., “Extending Requirements Analysis to Explore Alternatives”, *IEEE Software 18(1),* January/February 2001, 92-96.

[Poet] http://www.poet.com/products/cms/cms.html.

[Sahuquet00] Sahuquet, A., “Kweelt, The Making Of: The Mistakes Made and The Lessons Learned”, Technical Report Department of Information and Computer Science, University of Pennsylvania, 2000.

[Schmidt00] Schmidt, A., Kersten, M., Windhouwer, M., Wass, F., “Efficient Relational Storage and Retrieval of XML Documents”, Proceedings of the Third International Workshop on the Web and Databases, 2000.

[Shilakes98] Shilakes, C., and Tylman, J., "Enterprise Information Portals," Merrill Lynch, November 16, 1998.

[Simeon00] Fernandez, M., Simeon, J., Wadler, P., “XML query languages: Experiences and exemplars”, available at: http://www.research.att.com/~mff/files/exemplars.ps.

[Verity] http://www.verity.com/products/docnav/index.html.

[XSL] Extensible Stylesheet Language (XSL) version 1.0 – W3C working draft. Available at http://www.w3.org/TR/xsl/

[Yang99] Yang, Y., and Liu, X., "A Re-Examination of Text Categorization Methods," Proceedings ACM SIGIR Conference on Research and Development in Information Retrieval, 1999, 42--49.

[Yu93] Yu, E., "Modeling Organizations for Information Systems Requirements Engineering," Proceedings First IEEE International Symposium on Requirements Engineering, San Jose, January 1993, 34-41.

[Yu95] Yu, E., *Modelling Strategic Relationships for Process Reengineering*, Ph.D. thesis, Department of Computer Science, University of Toronto, 1995.

[Yu96] Yu, E., and Mylopoulos, J., "Using Goals, Rules, and Methods to Support Reasoning in Business Process Reengineering", *International Journal of Intelligent Systems in Accounting, Finance and Management 5(1),* January 1996.