Web Search with Metadata Links and Multimedia Presentations

G. Ozsoyoglu, M. R. Anderson, Z. M. Ozsoyoglu
Department of Electrical Engineering and Computer Science
Case Western Reserve University
Cleveland, OH 44106
tekin@eecs.cwru.edu, mra@po.cwru.edu, ozsoy@eecs.cwru.edu

6/16/00

Abstract

In this paper, we propose adding metadata to web for the purpose of performing semantics-based web searches and for producing multimedia presentations as a response to users' requests. The model uses (a) topics and "topic maps" as metadata in reaching to relevant worldwide web documents, (b) topic prerequisites and corequisites, also called *topic metalinks*, to define metadata-based navigational pathways on the web and to reach to web documents that constitute *sources,* and (c) multimedia presentations, called *Topic Comprehension Presentations,* for presenting possibly large sources to users in a controlled manner.

We investigate automated searching techniques that utilize in an integrated manner (i) topic metalink- and source-information in web documents and in topic map databases, (ii) web-accessible topic map databases, and (iii) local "user-profile" databases that contain users' knowledge on topics. The query output of such a search will be a list of Topic Comprehension Presentations, from which the user for playout purposes will choose one.

We describe the first version of a *Topic Comprehension Tool* that partially implements the proposed framework, and report its preliminary testing.

Contact Author: Gultekin Ozsoyoglu
tekin@eecs.cwru.edu
http://nashua.cwru.edu/
(216) 368-5029

Web Search with Metadata Links and Multimedia Presentations

G. Ozsoyoglu, M. R. Anderson, Z. M. Ozsoyoglu
Department of Electrical Engineering and Computer Science
Case Western Reserve University
Cleveland, OH 44106
tekin@eecs.cwru.edu, mra@po.cwru.edu, ozsoy@eecs.cwru.edu

6/16/00

Abstract

In this paper, we propose adding metadata to web for the purpose of performing semantics-based web searches and for producing multimedia presentations as a response to users' requests. The model uses (a) topics and "topic maps" as metadata in reaching to relevant worldwide web documents, (b) topic prerequisites and corequisites, also called *topic metalinks*, to define metadata-based navigational pathways on the web and to reach to web documents that constitute *sources,* and (c) multimedia presentations, called *Topic Comprehension Presentations,* for presenting possibly large sources to users in a controlled manner.

We investigate automated searching techniques that utilize in an integrated manner (i) topic metalink- and source-information in web documents and in topic map databases, (ii) web-accessible topic map databases, and (iii) local "user-profile" databases that contain users' knowledge on topics. The query output of such a search will be a list of Topic Comprehension Presentations, from which the user for playout purposes will choose one.

We describe the first version of a *Topic Comprehension Tool* that partially implements the proposed framework, and report its preliminary testing.

1 Introduction

HTML documents on the web specify display- or playout-only arrangement of text, audio, audio/video (A/V), image data, etc. With the exception of few metadata-tags such as <author> and <title>, HTML tags do not describe the content semantics of HTML documents. That is, HTML document designers have no standard ways of adding semantics to HTML data.

XML, the recently introduced web markup language [XML98], allows web document designers to specify the semantics of data in XML documents. For example, an XML document, using element and attribute names, may provide a video and text description of index mutual funds, as well as data needed for an analysis of the existing indexed mutual funds. Thus, XML introduces ways for adding semantics to web data.

In a recent work [EB00], we have used the notion of topics for modeling bulk text data (in "electronic books") for learning from electronic books. The notions of topics and topic hierarchies, in response to key-based searches, are currently being used to categorize the output of web search engines, e.g., Yahoo!. And, a recent industry-led effort proposes the notion of topic as metadata in describing data: the *Topic Maps* standard (ISO/IEC 13250) [Pe99, Bi99] is an effort to describe a metadata model for describing data in terms of topics, topic associations, and topic roles, and other topic-related metadata. Thus, as an example, one can attach a list of topics to bulk text or video data[1] as metadata. Topic Maps are also referred to as an example of "superimposed information" [MD99], and viewed as useful for variety of purposes.

We also think that searching or querying web data should be coupled with efforts designed toward comprehending the retrieved web data—in terms of what users already know and at what level, and what they need to know as prerequisites and corequisites. The recent use of topics and topic hierarchies by web search engines is a good step in the right direction, but it can be improved. Thus, in this paper, we propose to

[1] We anticipate that, in the near future, there will be standardized topic sets for different domains, similar in concept to namespaces for XML element names [NS99], or the electronic resource element set of the Dublin Core [DC95], or the container architecture of the Warwick framework [WF96].

investigate (a) topics as metadata for tagging and accessing web documents (XML/HTML), and (b) automated searching techniques (for obtaining topic information from the web) that utilize in an integrated manner (i) topic prerequisite and corequisite information in web documents, (ii) web-based topic map databases, and (iii) local "user-profile" databases that contain users' knowledge on topics. And, the search output of such a search will not be the URLs of a list of ranked web documents, but length- (i.e., duration) and height- (i.e., number of concurrent streams) controlled multimedia presentations, called Topic Comprehension Presentations or TCPs for short.

This paper addresses the following issues.

(a) Add new semantics and metadata information to web data in terms of
   (i) topics that characterize web data,
   (ii) topic metalinks that relate topics as prerequisites and corequisites,
   (iii) topic hyperlinks to sources as X-Pointer-like [XPTR99] references to parts of other web documents that are "relevant" to given topics, and
   (iv) topic map databases as full-fledged databases, which contain a topic data model, together with occurrences, types, contexts, and associations.

(b) Investigate automated ways for prerequisite-corequisite-based search of sources, and construction of the relevant TCPs.

1.1 Motivation for Extensions to Web Documents and Web Search Output

We propose the following extensions to web data and the associated web search, and the reasons for these extensions:

(a) Adding metadata to bulk text and multimedia data in XML documents, and creating secondary, invisible, stable, document-independent and compartmentalized navigational pathways on the web.

- Presently, users of HTML-based sites depend on each site's way of describing the text/multimedia data at that site. Quite often, such descriptions do not match to what the user is looking for. Or, the information that the user is looking for is available at the site, but hard to find (due to a large number of pages).
- For XML-based sites, from one site to another, there is no uniform way of telling what the site is describing: XML element and attribute names, while very useful in describing data in a given document, vary from one document to another[2]. And, the semantics of bulk text data in XML documents is not described in a standard way, and thus may not be useful to other sites.
- Information at an XML site (with its display (XSL [XSL98, XSLT99]) information) or at an HTML site can be better understood when presented with relevant information at other sites, but there is no *standard* way of knowing which other sites need to be looked at in order to comprehend the information specified at the current site. The web can benefit from tools that allow users to pose requests such as

  "collect information on topics *MP3* and *Digital Audio Formats Transformable to MP3* from 100 most relevant (i.e., reachable through topic metalinks) web sites, and prepare a basic-level, text-and-video-only multimedia presentation on what you have gathered, with text less than 5,000 words and video less than 20 minutes".

Presently, web users looking for information on a given topic surf, browse, and go through an arbitrary number of sites before locating what they are looking for. Sometimes, it is indeed a frustrating experience.

Topic maps [Pe99, Bi99] are proposed as a metadata model for use in the web for document indexing, and possible navigational purposes. We propose the use of topics and topic maps as metadata[3] in reaching to

[2] The recently introduced "XML namespaces" [NS99] attach "domains" to attribute names, and, thus, is an effort to reduce this problem.

[3] Resource Description Framework (RDF) [LR99], and the associated RDF Schema [LRa99] specify a foundation for processing metadata, and are recommendations from W3C. As such, our proposal may be viewed as an application of the RDF framework for a specific use.

relevant documents for the purpose of collecting information about a given topic or a set of topics, and delivering it to users in a controlled manner. We introduce and formalize notions such as *prerequisite, corequisite*, and *related topics* for accessing other relevant web documents. In the rest of the paper, we call these references as *topic metalinks,* which are to allow web designers to define metadata-based navigational pathways on the web. For example, assume that (a) the topic *Basic Digital Audio Formats* is a prerequisite to the topic *MP3*, (b) according to the user profile database, the user knows nothing about digital audio, and (c) the user requests information about the topic *MP3*. Then, the system puts together information not only about *MP3*, but also about *Basic Digital Audio Formats*.

Topic metalinks will be

- *Secondary* to the primary link mechanism on the web, i.e., hyperlinks.
- *Invisible* to users, and thus will not contribute to the information overloading suffered by web users.
- *Stable* in that, once defined (perhaps by a consensus over a group of domain experts), they will rarely change.
- *Document-independent*[4], as such information will be expressed between topics, not between web documents. And, topics will have contexts, providing *compartmentalization* of the associated metalinks. For example, the database context (where database is itself a topic) will contain topics relational model, OQL, etc.

To summarize, navigational pathways through topic metalinks will be complementary to the navigational pathways through hyperlinks.

(b) Presenting Data Accessed by Topic Metalinks: Topic Comprehension Presentations.

While reaching to relevant parts of web documents that contain bulk text and multimedia data is significant, just as important is how to present these possibly large amounts of data to users in a controlled manner for comprehension purposes. We think that multimedia presentations, concurrent and synchronized presentations of multimedia data, should be recast into this environment. We call such presentations *Topic Comprehension Presentations,* or TCPs for short.

(c) Prerequisite-Corequisite-based search of relevant XML documents through topics and sources.

Given a topic t by a user, the system (if directed) can locate a set T of other topics that are prerequisites and/or corequisites of t (possibly transitively) using topic metalinks. And, using hyperlinks to "occurrences" of topics (i.e., sources) in T, one can reach to bulk text/video/etc. data that describe topics in T. These are search requests as there is no query language—the user specifies a topic(s), and the system chases prerequisite, corequisite, or related topic information to reach to relevant topics and sources. Note that, unlike the search engines of today that return complete web documents, this approach will return, using XML tags and Xpointer-style pointers, *only* parts of web documents that describe the requested topics. Moreover, this approach will return *only* the "relevant" web data where relevancy is defined through the existence of a path of topic metalinks.

Thus, we propose *automated (reported in this paper) or query-based (not reported here) construction of multimedia presentations* (*in the form of TCPs*) *from both web documents and databases, in order for users to access topic-related information.* Our goal is to develop a framework for *Topic-Based Comprehension Tools* for web-based (text as well as multimedia) data.

Note that the primary use of the notions of topics and topic metalinks is for constructing TCPs and delivering information in a filtered and more useful manner—although such metalinks can certainly be used to enhance searching and browsing on the web. We view each of the comprehension tools, browsing tools, and searching tools as distinct and different in their primary goals:

[4] We think that stability and document-independence are important concepts in the Internet where the web information continually and quickly changes, and sites that are available today disappear tomorrow.

(i) *Comprehension Tools,* as we define, search *and* present web information through a set of prerequisite, corequisite, and related topics, and users' knowledge about topics. In the present web, this notion is lacking, and would be useful.
(ii) *Browsing Tools* are for moving around/surfing with the goals of picking up ideas, topics, etc. about information on the web. Browsing is ad hoc, undirected in nature, and not goal-directed.
(iii) *Searching Tools,* as defined by today's search engines, use Information Retrieval-based models for locating web pages from keywords/phrases, string matching, common hits, etc.

One can use each of the above tools towards the goals of the other since they are complementary and orthogonal to each other and since the web commonly contains incomplete and semistructured information. For example, to comprehend a certain topic, one can chase the topic metalinks of that topic either from a given set of XML documents in a database and/or on the web, or from a topic map database. However, given the incomplete nature of the web, not all of the relevant documents may be reached. Thus, a second possible step may be a complementary search using the specified topic as a keyword and a web search engine, with the goal of reaching to additional sources, and continuing with additional prerequisite/corequisite metalinks, etc.

Section 2 is the related work. In section 3, we summarize the Topic Maps standard. Section 4 describes our proposed data model for XML documents with topics and topic metalinks. In section 5, we discuss how to compute the closures of topics and topic metalinks. Section 6 lists some very basic web search requests and their time complexities. In section 7, we report on the first implemented version of the Topic Comprehension Tool.

2 Related Work

Extensible Markup language (XML) [XML98] is becoming a universal standard for data exchange on the web, recommended by the W3C Consortium. XML-Data [XMLD98] describes data in a self-describing format, either only through tags for elements and attributes (i.e., well-defined documents), or through separately defined schema (i.e., DTDs and valid documents).

Until recently, there has not been any standardization effort for multimedia presentations, which has resulted in a large number of complex, script-based commercial multimedia authoring tools [Auth96, Icon98, Quest98]. The Synchronized Multimedia Integration Language (SMIL) [SMIL98] is a recommendation from W3C for standardizing multimedia presentation specifications.

There have been many efforts for describing the contents of document sources on the Internet with the goals of providing support for the search, assessment and acquisition of information [GCMP97]. Within this context, an Internet source can be viewed as a collection of text documents and the associated search engine that accepts queries (of its own query language) from users, and produces ranked lists of documents. There are also metasearchers that, given a query about documents, select resources (with possibly different search engines and query languages), and transform the query into the queries of the selected sources, and send and have the transformed queries executed. Clearly, searching and resource discovery on the Internet is a very important problem, and our approach can be characterized as adding new "attributes" to source (text/multimedia) documents, and defining topic comprehension tools around these new attributes. Other work on defining attribute sets for bulk text data (i.e., documents) include Z39.50 Bib-1 attribute set [Bib1.95], the Dublin Core [DC95], the Warwick Framework [WF96], and Basic-1 attributes of STARTS [GCMP97]. The source metadata attribute sets include Z39.50 Exp-1 attribute set [Z3950], the GILS profile [GILS99], and Mbasic-1 attribute set [GCMP97].

3 Topic Maps Standard

We now briefly summarize the Topic Map data model, as described in [Pe99, Bi99]. The standard *ISO/IEC 13250:1999 Topic Maps* describes a model and declares an interchange format for topic maps. A topic map organizes large sets of information resources by building a structured network of semantic links over the resources [Rath99].

A *topic map* is an SGML (or XML) document (perhaps a structure, a file or a database) that contains a topic data model, together with occurrences, types, contexts, and associations. The essential concepts are the following [Rath99]:

- *Topic:* Definition of a topic is very general: a topic can be anything about which anything can be asserted by any means. As an example, in the context of Encyclopedia, the country Spain, or the city Rome are topics (about subjects).
- *Topic type:* Topics are *typed,* (e.g., type of the topic Rome is city), and have names. Topic names have *scopes,* e.g., language, style, domain, etc. Topic names are also typed; e.g., base name (required), display name (optional), etc.
- *Source:* Topics have *occurrences,* which we call *sources* in this paper*,* within addressable information resources. For example, a topic can be *described* in a monograph, *depicted* in a video or a picture, or *mentioned* in the context of something else.
- *Topic association:* An association describes the relationship between two or more topics. For example, topic Rome *is-in* topic Italy; topic Tom Robbins *was-born-in* topic USA, etc. Each association is of a specific association type. Each associated topic plays a role in the association. The association type and association role type are both topics.
- *Scope and theme:* Any assignment to a topic is considered valid within certain limits, which may or may not be specified explicitly. The validity limit of such an assignment is called its scope, which is defined in terms of topics called themes.

An important aspect of topic maps is that topic associations are *completely independent* of whatever information resources may or may not exist as occurrences of those topics. This means a topic map is great source of information in its own right. A topic map, rather than the actual occurrences of the topics, reveals the organization of knowledge. This allows a separation of information into two domains: the topic domain and the occurrence (document) domain. Computations can be carried out on the topic domain without regard for the sources. Once the computations have been completed, the results can be translated into links to the document domain.

In addition to the topic map standard, the ISO working group has introduced the term *topic map template*. This term is currently "semi-official", since it has not yet been included in the standard. A topic map template is itself a topic map consisting of all the constructs that have declarative meaning for another topic map. It is essentially a declaration of topic types, occurrence role types, association types, and other items that form the basic elements of a given topic map.

A template can be copied into or referenced by other topic maps. It can also be used as a starting point for a new map that can be extended during further development of the map. Topic map templates can be merged with one another, allowing for a great deal of extensibility in topic map design. From the descriptions above, a topic database data model is quite similar to the Entity-Relationship model specialized for the abstract domain of topics and topic-related information.

4 Data Model for XML Documents with Topics and Topic Metalinks

We now describe the data model of our environment that contains XML documents, web-based topic map databases, and local (user-application-accessible) user profile databases.

- *Topic detail levels*.

  For each topic, we assume that there are a number of integer-valued[5] *topic detail levels* describing how advance the level of the topic is. To illustrate topic detail levels, for example, the knowledge of a user on

[5] Detail levels may be textual classifications such as *beginner, intermediate, advanced, expert,* etc. For simplicity, here, we use integers.

the topic *Stock Market* can be at a beginner (i.e., detail level 1) level, e.g., only *Basic Investing* and Stock *Market Indexes*. Or, it may be at an advanced (say, detail level n) level, e.g., *Risk Analysis and Random Walk Theory*, etc. Topic x at detail level i is more advanced (i.e., more detailed) than topic x at detail level j when i>j. We also assume that topics are organized into *Topic Hierarchies* (TP hierarchy) within different contexts. An example to topic hierarchies is hierarchically organized keywords (i.e., topics) at the end of hardcopy books.

- *Web-based Topic Map databases.*
  A topic map database contains an instance of a topic map data model, together with hyperlinks or Xpointers to XML documents for sources containing bulk text/video, etc. data. Clearly, sources will be document-dependent, and, as the web changes, they change as well or become *invalid* sources.

- *User-application-accessible user profile databases.*
  A user profile database contains the *knowledge levels of users on topics.* For a given user and a topic, the knowledge level of the user on the topic (zero, originally) is a certain detail level of that topic, and is kept in the user's profile. We will use this information for the automated construction of Topic Comprehension Presentations[6] (TCPs). In the absence of a user profile, the user is assumed to know nothing about any topic, i.e., the user's knowledge level about all topics is zero. The user profile database description given here is very preliminary, and, it is likely that, in different application domains, different and more detailed models of "user knowledge" may be needed.

- *Topic prerequisites and corequisites as metalinks.* Some hardcopy textbooks (e.g., see [Sav96]) provide "*dependency diagrams"* in an attempt to help instructors/students choose the order of topic coverage. We adapt this notion into the web environment. For example, the prerequisite to discussing the topic *Mutual Fund Costs* in the *Financial Investments* domain is the coverage of the topics *Loading Fees* and *Expense Charges.* For electronic books [EB00], we formalized this concept into the concept of prerequisite dependencies among topics, and used it for electronic book databases, for automated "lesson" construction from electronic books. We now revisit the same notions for the web environment as *topic metalinks*[7]. For example, for the *Funds* topic domain, we may have the prerequisite constraint specified as "the topic *Loading Fees* (*l*) should be described after understanding the topic *Mutual Funds* (*m*)". That is, the prerequisite metalink $l \rightarrow m$ holds. If topic y is a prerequisite to topic x for the comprehension of topic x, (i.e., x→y holds), when a user on the web requests a topic comprehension presentation (TCP) on topic x, the system makes sure that topic y is "covered" first, and topic x is covered next. We require that, when a user requests a TCP on x, and his user profile knowledge level on y is zero then the constructed TCP should also have a source for y in the form of text/video/etc. stream.

  The prerequisite notion specified in the above paragraph constitutes the most basic form of prerequisite specification. One can also investigate other forms of prerequisites such as "the prerequisite to comprehending topic x is to comprehend at least n number of "examples" from the occurrences of another set S of topics".

  Please note that we actually use prerequisite metalinks among topics at different detail levels, e.g., the prerequisite metalink $l^4 \rightarrow m^1$ states, "the prerequisite to *Loading Fees* at detail level 4 is the knowledge of *Mutual Funds* at detail level 1 *or higher*". As an illustration, consider the interpretation of the two prerequisite metalinks shown in Figure 1.

[6] We will not deal with the important issue of how the knowledge level of a user is determined, as it is not in our domain of expertise. This issue is discussed in our work on electronic books [EB00]. We will assume that, through some means (e.g., tests, quizzes, etc.), the knowledge level of a user on a topic is established.

[7] We have chosen to use the terminology *topic metalinks,* as opposed to *topic dependencies* in our work on electronic books, in order to emphasize that a topic metalink is *meta*data, and forms a type of navigational *link*.

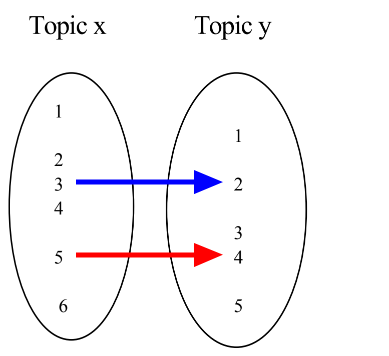


Figure 1. Prerequisite Topic Metalinks

(1) If topic x at level 3 is presented in a TCP, say S, then topic y at level 2 or higher must either (i) be in S, or (ii) is already in the user's profile. That is, the prerequisite metalink $x^3 \rightarrow y^2$ holds. Please note that, in this prerequisite metalink, the right hand side specifies "topic y at level 2 or higher", as opposed to "topic y at level 2".

(2) If topic x at level 5 is presented in a TCP, say S, then topic y at level 4 or higher must either (i) be in S, or (ii) is already in the user's profile. That is, the prerequisite metalink $x^5 \rightarrow y^4$ holds.

We assume that topic metalinks are dynamically or statically collected from XML documents in the web or in databases, and made available in web-accessible topic map databases. And, document designers may also specify topic metalinks in XML documents. Similar to prerequisite metalinks, we also have the notions of (a) *topic corequisites*, specifying the corequisites of the given topic, and (b) *related topics* for a given topic that are neither prerequisites, nor corequisites, but are simply related.

Given a topic t, topic metalinks of t can be represented as a *topic precedence graph*[8]. Figure 2 illustrates a topic precedence graph of t where the edge labels *pre* and *co* identify prerequisite and corequisite information, respectively. For example, topics g and e are among the prerequisite and corequisite topics of t, respectively.

- *Topic-description tags within web (XML) documents to specify sources (topic occurrences).*
  We assume that web documents with bulk text/video/still images, etc., carry topic "tags", stating for, say, bulk text data,
  (a) the topics that the data "describes" (e.g., <topic-description>, </topic-description> tags),
  (b) the detail level of the description, and
  (c) pointers that indicate where the description starts and where it ends, specified perhaps through a mechanism such as Xpointers [XPTR99]. Topic description tags define *sources* (topic occurrences) for topics.
- *Topic hyperlinks to topic-description tags (sources) in other web documents.*
  Topic hyperlinks contain two pieces of information: they specify both a topic name with its detail level, and an actual hyperlink to the topic description tags of bulk text/video/etc. data (i) from a given web (XML) document, or (ii) from a topic map file.
- *Topic Comprehension Presentations (TCPs).*
  Users request TCPs about topics. Each TCP is a multimedia presentation whose nodes are topic hyperlinks to sources.

[8] We use the graph representation here for the sake of illustration. Graph representation of topics is not always sufficient to capture metalink "closures", which is discussed next in section 5.

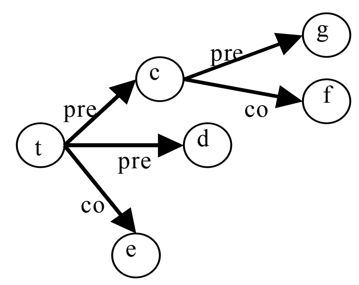


Figure 2. Topic Precedence Graph of Topic t

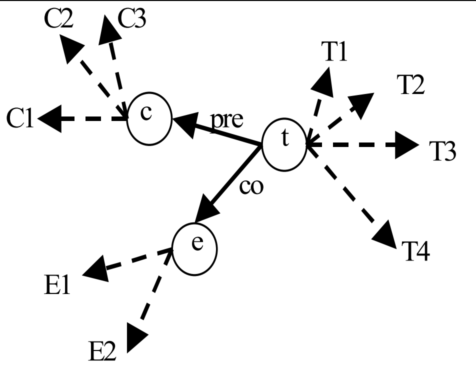
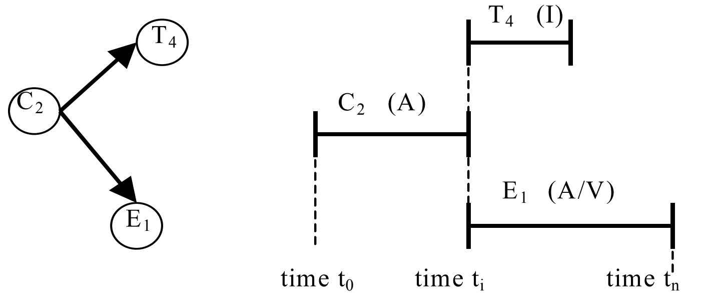


Figure 3. Multiple Topic Occurrences for Topics

Example. From Figure 3, a TCP for topic t may contain sources C_2 of topic c, E_1 of topic e, and T_4 of topic t, ordered such that the playout of C_2 precedes the playout T_4 (since, from Figure 2, c is a prerequisite of t), and playouts of E_1 and T_4 proceed concurrently (since, from Figure 2, e is a corequisite of t). Figure 4 below contains two representations of the resulting TCP, namely, Figure 4(a) as a directed acyclic graph where each node represents a source as a multimedia stream, and each edge represents the order of playout, and Figure 4(b) as a timeline graph where the horizontal axis represents the time duration $[t_0, t_n]$ of the overall presentation, and the vertical axis represents the sources. The sources C_2, T_4, and E_1 are multimedia streams of type Audio, Image, and Audio/Video, respectively.



(a) TCP as a DAG (b)TCP as a Timeline Graph

Figure 4. An Example TCP

We define the *length of a TCP* as the presentation time duration (e.g., the length of the TCP in Figure 4 is $t_n - t_0$), and the *height of a TCP* as the maximum number of concurrently played streams at any given time during its playout (i.e., the maximum vertical "cut" of the timeline graph. The height of the TCP in Figure 4 is 2).

In addition to time length, it is quite natural for users (or for the system) to impose limits on the height of a TCP to be generated. For example, the users may request that the TCP to be generated should not have more than two concurrently played out video streams.

A source for a topic can be a bulk text in one document, a video stream in another document, and a still picture in yet another document. Therefore, when a TCP about a topic is requested, depending on the time constraints given by the user, a source of one type may be shorter in time, and a better choice than others. On the other hand, there are also performance issues in choosing a source from a set of sources in that downloading to the user's site a large (e.g., video) source for playout purposes may be more costly than downloading a text-only source. Yet another requirement is not to concurrently schedule more than one audio (or, audio-video) streams as this is physically not possible. Clearly, user-defined rules can be used as to which and how many of these different types of sources describing the topic at hand should be included into a TCP. Thus, one can investigate different models and optimization issues on how to choose from a set of sources for inclusion into a TCP.

- *Topic coverage in a TCP.*
  We say that the TCP S *covers* topic t at level i if S contains in it (a hyperlink to) a source for the topic t at level i.

5 Closure Computation for Topics and Prerequisite Metalinks

For the sake of simplicity, in this section, we only discuss prerequisite topic metalinks, and ignore corequisite and related-topic metalinks. Therefore, in this section, topic metalink, or simply metalink, refers only to a prerequisite topic metalink.

One can use four different models for representing prerequisite topic metalinks. The semantic differences between the models may be due to whether or not prerequisite topic metalinks are allowed to be

a) *cyclic* (e.g., $x \rightarrow x$ forms a trivial cycle; $x \rightarrow y$ and $y \rightarrow x$ form a non-trivial cycle). The alternative is to allow only *acyclic* prerequisite topic metalinks, e.g., trivial or non-trivial cycles are not allowed.
b) (left-hand-side) *decomposable* (e.g., $xy \rightarrow z$ is equivalent to $x \rightarrow z$ and $y \rightarrow z$) or *nondecomposable* (e.g., $xy \rightarrow z$ is not equivalent to $x \rightarrow z$ and $y \rightarrow z$).

The simplest prerequisite model that is commonly used in hardcopy textbooks allows only acyclic and decomposable prerequisites. However, one can also have web environments in which prerequisite metalinks are cyclic and/or nondecomposable. Consider the case of a cycle of three metalinks, namely, $x \rightarrow y$, $y \rightarrow z$, $z \rightarrow x$ among topics x, y, and z. We can interpret the existence of this cycle as "in any topic comprehension presentation (TCP) request having one of topics x, y, or z, coverage of all three topics must be included into the constructed TCP". Clearly, this attaches a separate semantics to a cycle of prerequisite metalinks than the semantics of a single prerequisite metalink.

Let's consider decomposability. The prerequisite metalink $ab \rightarrow c$ states that " a and b together in a TCP formation request has c as the prerequisite" or "the prerequisite of a and b is c". We say that $ab \rightarrow c$ is nondecomposable if $ab \rightarrow c$ does not imply that $a \rightarrow c$ and $b \rightarrow c$. (Note that the reverse is always true, i.e., the prerequisite metalinks $a \rightarrow c$ and $b \rightarrow c$ always imply the prerequisite metalink $ab \rightarrow c$). Below we illustrate a case in which prerequisite metalinks are nondecomposable.

Example. Assume i, l, and b represent the topics *Indexed Mutual Funds*, *No-Load Mutual Funds*, and *Basics of Mutual Funds*, respectively. When a TCP about both *No-Load Mutual Funds* and *Indexed Mutual Funds*, both at level 2, is requested, it may make sense to include the topic *Basics of Mutual Funds* at level 1 into the TCP for completeness (thus, the prerequisite metalink $i^2l^2 \rightarrow b^1$). However, we may not require *Basics of Mutual Funds* to be included into the TCP if only one of *Indexed Mutual Funds* or *No-Load Mutual Funds* is requested (e.g., $i^2l^2 \rightarrow r^1$ is not equal to $i^2 \rightarrow b^1$ and $l^2 \rightarrow b^1$).

In the rest of this section, we discuss how to compute

(a) *topic closures* (i.e., given a set X of topics, obtaining the closure X^+; the set of topics that are prerequisites to topics in X and are not known by the user) and

(b) *prerequisite closures* (i.e., given a set P of prerequisites, the set P^+ of all the implied prerequisites) when prerequisite metalinks are cyclic/acyclic and decomposable/nondecomposable.

If prerequisite metalinks are nondecomposable and allowed to be cyclic then their semantics is equivalent to the semantics of functional dependencies. That is, prerequisite metalinks can be axiomatized using Armstrong's axioms, which are sound and complete [Ullm89]. One can then compute P^+, the closure (i.e., the set of implied prerequisite metalinks) of a set P of prerequisite metalinks. More interestingly, one can find the closure (i.e., all the prerequisite topics) X^+ of a set X of topics by using the O (N.L) closure algorithm for a set of attributes [Ullm89] where N is the number of prerequisite metalinks, and L is the length of the encoding for a prerequisite metalink.

If prerequisite metalinks are acyclic and decomposable then a given topic cannot be a prerequisite to itself. This means that the reflexivity axiom for functional dependencies does not apply to prerequisite metalinks of this model. Similarly, augmentation axiom of functional dependencies does not apply either[9]. Also, this model allows prerequisite metalinks of the form $xy \rightarrow z$ to be equivalent to $x \rightarrow z$ and $y \rightarrow z$, which is not true for functional dependencies. For this case, to find the closure P^+ of a set P of prerequisite metalinks, we can first "fully" decompose all prerequisite metalinks into P' so as to have only one topic in the left-hand-side and the right-hand-side of each metalink. Then, we can create a precedence graph G_P (V, E), where V is the set of topics, and the set E of edges contains the edge from node a to node b if and only if P' contains the prerequisite metalink $a \rightarrow b$. The closure P^+ of P can then be found by finding the transitive closure of G_P. And, the closure X^+ of a set of topics X can be found by locating all topics that contain nodes in G_P reachable from each of the nodes in X. Also note that we can check the acyclicity of a set of prerequisite metalinks in this model by simply checking the existence of a cycle in its precedence graph in linear time.

If prerequisite metalinks are cyclic and nondecomposable then finding the closure P^+ of a set P of prerequisite metalinks is identical to the solution in the above paragraph. We first "fully" decompose all prerequisite metalinks in P into P' so as to have only one topic in the left-hand-side and the right-hand-side of each metalink. Then, we create the precedence graph G_P(V,E), where V is the set of topics, and the set E of edges contains the edge from node a to node b if and only if P' contains the prerequisite metalink $a \rightarrow b$. The closure P^+ of P can be found by finding the transitive closure of G_P. And, the closure X^+ of a set of topics X can be found by finding all nodes in G_P reachable from each of the nodes in X.

If prerequisite metalinks are acyclic and nondecomposable then the left-hand-side of a prerequisite metalink may contain multiple topics. In this case, one may think of using a precedence graph where the node from which an edge emanates contains a set of topics. Such a graph leads to a hypergraph as a precedence graph. However, unlike the solutions in the above paragraphs, the transitive closure of such a graph would not capture all the metalinks. Consider, for example, the set of metalinks $\{x \rightarrow a, \; ab \rightarrow c\}$, and the request for the closure of the set {x, b} of topics. The transitive closure of the precedence graph returns {x, a, b} as the answer whereas the correct answer should be {x, a, b, c}. Thus, transitivity itself is not sufficient for topic closure. In the rest of this section, we specify a sound and complete axiomatization for this case, and also describe the related topic closure algorithm, both adapted from [EB00, EBa00]. For proofs and details, please see [EBa00].

We observe that Armstrong's Axioms, used to axiomatize standard functional dependencies are not appropriate when acyclicity is demanded. The axiom of reflexivity in Armstrong's axioms generates trivial (weak) cycles, as does the axiom of augmentation in Armstrong's axioms.

Def'n: Pseudo-transitivity axiom: If $X \rightarrow Y$ and $WY \rightarrow Z$ then $WX \rightarrow Z$.

Def'n: Split/join axiom: if $X \rightarrow AB$ then $X \rightarrow A$ and $X \rightarrow B$, and vice-versa.

[9] Given $x \rightarrow y$ and z, $zx \rightarrow zy$ is valid for functional dependencies. However, for prerequisite metalinks, when z is replaced by x, we have $xx \rightarrow xy$, which creates a trivial cycle and is not allowed.

Theorem 1: The pseudo-transitivity and split/join axioms are sound and complete.

The following algorithm, a slightly revised version of the attribute closure algorithm for functional dependencies [Ullm89], computes the closure of a set of topics X:

Algorithm:

1. $X^{(0)}$ is set to empty.

2. $X^{(i+1)}$ is $X^{(i)} \cup \{y\}$ such that there is a dependency in F of the form $X_i \rightarrow y$, where $X_i \subseteq X \cup X^{(i)}$ and $y \notin X$.

The algorithm terminates when $X^{(j)}=X^{(j+1)}$ (when no dependency can be invoked), and the output X^+ is $X^{(j)}$. Clearly it will always terminate.

Lemma 1: Algorithm 1 correctly computes X^+.

This algorithm is functionally equivalent to that described for functional dependencies in [Ullm89], and runs in time linear in the size of the representation of the dependencies.

Lemma 2: Computation of the closure of a set of topics X under a set F of acyclic nondecomposable dependencies does not violate acyclicity. That is, $X \quad X^+$ will not imply any cycles.

6 Topic-Based Web Search With a TCP as Output

As we have described in the introduction, a user can simply list a set of topics to be searched over the web through metalinks as well as a number of constraints (to be listed below), in which case the result to be returned is the "best" TCP that satisfies the user's request. Or, the user can pose a web query such that the evaluation of the query through topic metalinks returns a "set" of TCPs that satisfy the query. In this section, we will characterize the first approach, which we will call "*topic-based web search*" requests.

Below, we consider *linear TCPs*, which are TCPs with height one (i.e., at any given time during the playout of the TCP, there is *only* one multimedia stream that is being played out) as well as *nonlinear TCPs* which are arbitrary directed acyclic graphs. Clearly, the height of a linear TCP is one, and the height of a nonlinear TCP is greater than one. We also discuss *height-bound* TCPs where the user species an upper bound on the height of the TCP to be created. To simplify our discussion, we assume that for each topic, there are exactly n, $n \geq 1$, sources.

We now list two possible "constraints" in topic-based web search requests.

(a) *TCPs about topics with constraints on multimedia types.* An example request is "Given the set X of topics where each topic in X has a level specified, prepare a text-only TCP with at most 1000 words".

(b) *An upper bound t_{UB} on the length and height of the TCP*. An example is "prepare a video-only TCP of at most 30 minutes long on topic x, and the number of concurrent videos should at most be three".

We now briefly characterize some typical topic-based web search requests, and investigate their time complexities. This list is preliminary and presented as a sample. Depending on the chosen domain of topics, we envision that completely different sets of topic-based web search requests may make sense. For each of the requests below, we assume that we are given (a) the user's knowledge levels for topics (i.e., the user profile database), and (b) prerequisite metalinks in topic map databases, and in XML documents.

Web Search Request 1. Given a set X of topics, search the web and produce a TCP on topics X, in the order given, at the highest levels.

Remark 1: For linear or nonlinear TCPs, the web search request 1 can be satisfied in polynomial time in the number of topics in the closure of X.

The request 2 below adds a time bound (i.e., TCP length) constraint to request 1.

Web Search Request 2. Given (a) a set X of topics, and (b) a time bound t_{UB} on the length of the TCP to be produced, search the web and produce within the time bound t_{UB} a TCP that covers as many topics in X as possible, in the order given in X, at the highest levels.

Remark 2: For linear TCPs, the web search request 2 can be satisfied in polynomial time in the number of topics in the closure of X.

Remark 3: For nonlinear TCPs with a height bound m, $m > 1$, the web search request 2 is NP-complete.
Proof: By a polynomial reduction from the k-clique problem.

The request 3 below modifies request 2 with sources having variable detail levels.

Web Search Request 3. Given (a) a set X of topics, and (b) a time bound t_{UB} on the length of the TCP to be produced, search the web and produce a TCP that has a length of t_{UB} or less, and covers as many of the topics in X (at any level) as possible.

Remark 4: For linear or nonlinear TCPs, the web search request 3 is NP-Complete.
Proof: By a polynomial reduction from the k-clique problem.

Empirical evaluations of heuristic solutions for the above-listed web search problems will be done when we complete the implementation of the Topic Comprehension tool, whose first version is described next.

7 The Topic Comprehension Tool, Version 1

We have implemented the first version of the Topic Comprehension Tool[10] [And00] that, at the present, dynamically locates and organizes topics on the basis of metalinks, and allows web-based searches. The metalinks we have presently implemented are:

(a) one topic **BuildsOn** (i.e., is a prerequisite to) another,
(b) a topic **LeadsTo** (i.e., is a postrequisite to; converse of prerequisite) another, and
(c) a topic **Complements** (i.e., is a corequisite to) some other topic

In our current implementation, the Topic Comprehension Tool, when used in conjunction with topic sources (i.e., topic occurrences) allows for the construction of *a list of suggested readings* for learning a specific set of topics. We are presently extending the tool [Gao00] to construct the associated linear and nonlinear TCPs for web search requests. In the near future, we plan to add user profile databases and preference mechanisms into the tool.

The Topic Comprehension Tool analyzes documents for special XML markups of topics as well as topic metalinks, and then stores these topics and metalinks in a topic map file. The tool offers real-time searches.

[10] This first version was originally called the "Topic Browser" Tool.

That is, as a document is being read, it is semantically analyzed, and its topics are related to topics contained in other documents. Those documents, linked to the original document by topic hyperlinks, are then visited by the Topic Comprehension Tool and treated similarly. From this process, a set of related topics is found and each topic is associated with a set of sources. By computing the closure of the topic metalinks (presently implemented only as acyclic and decomposable metalinks) and finding each source associated with each topic, a set of semantically related sources is created. This is then used to create *a suggestion list*. The suggestion list as computed by the Topic Comprehension Tool can take a number of forms. The closure calculation on the graph of metalink-related documents produces a set of related topics and a set of sources describing those topics. From these sets, various views of the data can be created to display the information in a manner requested by the user.

The Topic Comprehension topic map is a data structure that contains topics and topic sources. The topics and their sources can be considered to exist in separate domains within the topic map. Because of this, it is possible to perform calculations on topics without needing to consider the sources in which those topics appear. Following the calculation, the sources of the topics can then be examined.

A number of different source types may exist. These include informational documents, tests, and experiment suggestions. Informational documents provide information on any number of topics. The goal of reading one of these documents is to learn about the topics contained within it. A test document provides a number of questions or problems about various topics to allow the reader to test his knowledge. An experiment suggestion document describes experiments that could be performed to allow the reader to gain knowledge on a topic through experimentation (i.e., “hands-on” learning). The suggestion list provided to the user by the Topic Comprehension Tool could differentiate between the document types and provide hyperlinks to those types desired by the user.

7.1 Implementation

The Topic Comprehension Tool is presently implemented using a three-tiered architecture, dividing the system into three distinct parts: the presentation layer, the application layer, and the data layer. The Topic Comprehension Tool is written in Java, using the Sun Java Servlet engine and web server. XML parsing is accomplished using the Java XML parser developed by the World Wide Web Consortium (W3C).

7.1.1 Presentation Layer

The presentation layer consists of a web browser and a web server. A user accesses the main Topic Comprehension Tool web page and enters either a URL of a source or a list of topic names. Entering a URL brings up a page listing the topics in the source. Entering a list of topic names brings up a page listing the topics and the topics associated to each topic in the list.

From the page generated by the topic list, the user selects topics upon which to base a closure computation. There are also configuration options available for the user to select, including the number of sources to return and the metalink type to use to build the suggestion list. This information is sent in a request to the web server. Upon completing the processing of the request, the Topic Comprehension Tool web server returns a web page displaying the topics found by the closure calculation and the source in which they appear.

Users can specify a number of parameters that will act as filters on the suggestion list data. By giving a detail level range, the user can limit the sources listed to those that fall within that range. A maximum number of sources per topic can be given.

7.1.2 Application Layer

7.1.2.1 Dynamic Topic Map Creation

The Topic Comprehension topic map is a modified Java Hashtable object that stores Topic and Source objects, both of which are MapItem objects. Topics are hashed using the topic name as a key, while Sources are hashed using the source’s URL.

The dynamic creation of a topic map is initiated by a user's request for a suggestion list. A user visits the Topic Comprehension Tool web page and enters either a topic name or a document's URL upon which the suggestion list will be based. If a topic name is entered, the global topic map is searched for the topic and will be used for creation of the suggestion list if the topic is found. If a URL is entered, the Topic Comprehension Tool will visit the document and the global topic map will absorb the topics marked up in the document. The Topic Comprehension Tool visits a document and finds the <TopicBrowserMarkup> tags, which contain descriptions of the topics and their associated topics. The topic map creation process now begins:

As the markup for each topic contained in the document is processed, a Topic object is created. This object contains pointers to each associated topic, each source containing the topic, and to the TopicMap object to which it belongs. In the topic's markup are the markups of other topics, each associated to the main topic. The topic creation is done recursively; as the associated topic markups are encountered, Topic objects are created, associated with the main topic, and added to a TopicMap object for the Source being processed. The metalinks are reciprocated automatically as they are created (i.e., if the main topic **BuildsOn** the associated topic, a **LeadsTo** metalink is added to the associated topic pointing to the main topic).

Since different experts could have authored the sources being examined by the Topic Comprehension Tool, topic metalinks may differ between authors. One author may think *A* **Complements** *B*, while another may think *A* **BuildsOn** *B*. Presently, such conflicts are resolved by what is essentially a vote. A tally is kept of the number of occurrences of each association for each topic. The metalink with the most occurrences is used in the suggestion list. We are currently investigating more advanced mechanisms for "merging" different author's preferences, along the lines of [AgW00].

Once each topic has been created, the topic's sources are processed. A Source object is created for each one that is marked-up. A Source object points to each Topic object the source contains and to the TopicMap object to which it belongs. For each Topic pointed to by the Source object, a DetailLevel object is stored. The DetailLevel object describes the detail level of the topic in the source. The detail level is a real number between 0 and 10, where level 10 is the highest detail level possible. The creator of the source judges the source's detail level for a particular topic and marks it in the source tag. The other sources marked up in the document can also be given detail levels. The DetailLevel object takes an average of all the detail levels encountered for a particular topic in a particular source returns that as the detail level for that topic.

Once the processing of a Topic has been completed (all of its associated Topics and Sources have been created), the Topic's TopicMap object is merged with the TopicMaps of its associated Topics. Duplicate Topics are merged, meaning the topic metalinks and sources for each topic are combined and all pointers to each topic are set to point to the merged topic. Duplicate Sources are treated similarly.

After all the merging has been performed, the TopicMap for the original source is merged with the global TopicMap object, which exists if previous suggestion lists have been created.

7.1.3 Suggestion List Creation

A ClosureCalculator object processes the global TopicMap object to create a suggestion list. The user specifies the type of list to create (**BuildsOn, Complements,** or **LeadsTo**). Starting with each topic in the original source (or a specific topic if given by the user), the ListMaker traverses the proper topic metalink in each Topic object. For the **BuildsOn** metalink, topics are displayed from farthest to closest from the original topics. For the **LeadsTo** and **Complements** metalinks, the topics are displayed from closest to farthest. The ordering was chosen to present the topics and sources in the order in which a user most likely would want to learn them. For each topic, the output is as follows:

Topic: *topic name*
Sources: *table of source names, URLs, and their detail levels*

The topic names are hyperlinks that link to the creation of a new suggestion list based solely on that topic. The source names are hyperlinks to that source.

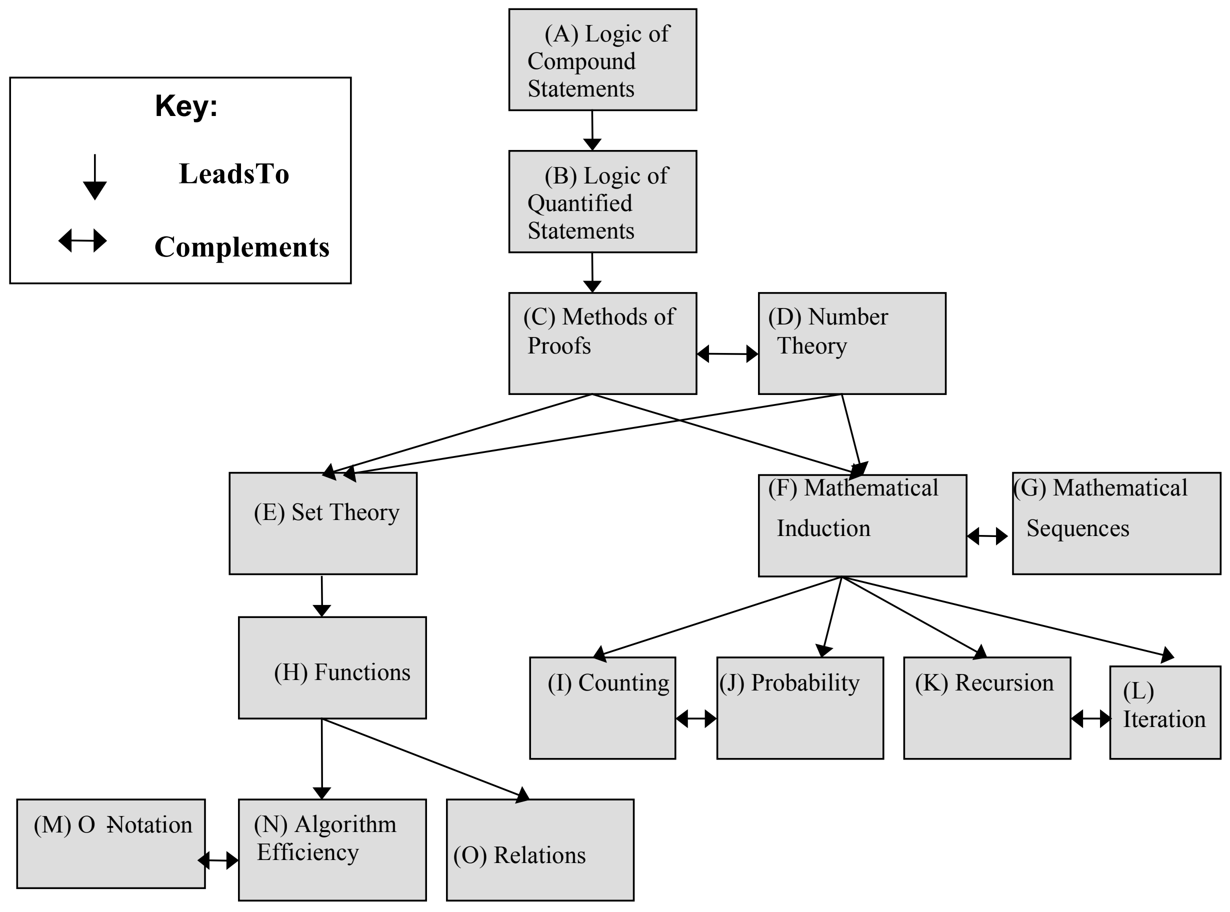


Figure 5. Topic Hierarchy Used in Testing

7.2 Data Layer

The data layer is the global topic map. By adding the topics and sources marked up in each source examined by the Topic Comprehension Tool, the number of topics and sources that can be examined for inclusion increases for each list created. The global TopicMap object stays in memory while the Topic Comprehension Tool server is running and is serialized to disk using the Java Serializable interface. Whenever the server is started, it checks for the existence of the serialized TopicMap. If it is found, it is loaded into memory. If not, the Topic Comprehension Tool creates a new, empty TopicMap.

7.3 Preliminary Testing of the Topic Comprehension Tool

A topic hierarchy, shown in Figure 5, was created based on the organizational tree diagram appearing in the Preface of Susanna S. Epp's book, *Discrete Mathematics with Applications* [Epp95]. The diagram shows only **LeadsTo** and **Complements** metalinks. From this hierarchy, ten sample sources were created so each contained some of the topics (see Table 1). XML documents marking up the topics and their metalinks were created for sources 1-7. No documents were created for sources 8, 9, and 10, allowing the behavior of the Topic Comprehension Tool to be tested when a document cannot be found.

Source	URL	Topics Contained	Sources Referenced
1	http://vorlon.cwru.edu/~mra/1.xml	A,B	2,4,10
2	http://www.angelfire.com/oh4/liverwort/2.xml	A	1,5,10
3	http://cosmicweb.tripod.com/xml/3.xml	D,E,F	4,5,7,10
4	http://vorlon.cwru.edu/~mra/4.xml	C,F,G	3,5,6,9,10
5	http://www.angelfire.com/oh4/liverwort/5.xml	D,F,K	3,4,6,9,10
6	http://cosmicweb.tripod.com/xml/6.xml	I,J	4,5,10
7	http://vorlon.cwru.edu/~mra/7.xml	H,O	3,7,8
8*	http://www.angelfire.com/oh4/liverwort/8.xml	M,N	n/a
9*	http://cosmicweb.tripod.com/xml/9.xml	K,L	n/a
10*	http://vorlon.cwru.edu/~mra/10.xml	A,B,D,E,F,G,H I,J,K,L,M,N,O	n/a

- Document does not actually exist

Table 1. Sources Used in Testing

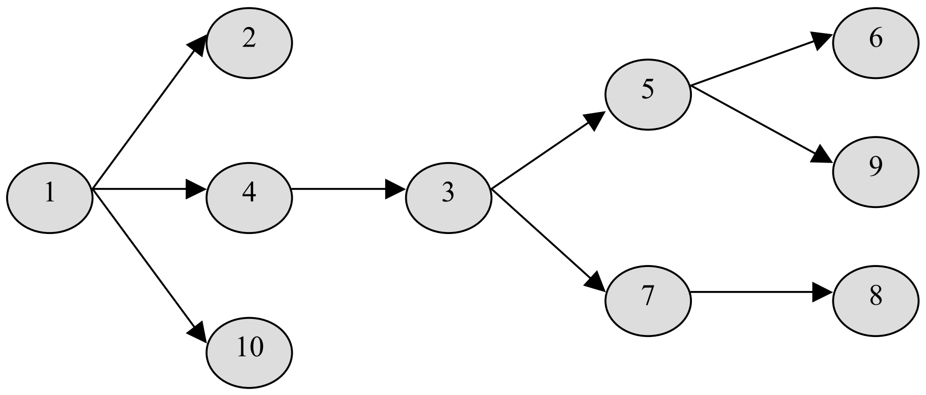

$$\begin{tikzcd}
 & 2 & & & 6 \\
 & & & 5 \arrow[ur] \arrow[dr] & \\
1 \arrow[uur] \arrow[r] \arrow[ddr] & 4 \arrow[r] & 3 \arrow[ur] \arrow[dr] & & 9 \\
 & & & 7 \arrow[r] & 8 \\
 & 10 & & &
\end{tikzcd}$$

Figure 6. Reaching to Sources When Starting the Search From Source 1

Average	354 ms
Shortest	110 ms
Longest	3190 ms
Std. Deviation	561 ms
Longest for 10 sources	7890 ms

Table 2. Access Times for Sources

The Topic Comprehension Tool was given the URL of one of the sources and then it built a topic map based on that source and the sources it could reach from references contained in it and in the documents referenced by it. For example, when given the URL of source 1, sources 2, 4, and 10 would be visited (or an attempt would be made to visit them, in the case of source 10). From those sources, sources 3, 5, 6, 9 would be visited, and from them, sources 7 and 8 would be visited (see Figure 6). Thus all sources would be visited and the entire topic map would be constructed. Not all sources are reachable from every source, though. When starting with sources 3-7, sources 1 and 2 cannot be reached. If sources 8, 9, or 10 were the starting points, no other sources could be reached, since those sources do not actually exist.

When a document cannot be reached, a Source object is created in the topic map for it anyway. This allows authors to mark-up sources that aren't available on the web currently or to mark-up sources whose web servers are temporarily down. Whenever the Topic Comprehension Tool accesses that Source object, a flag (isVisited) is checked for the Source. If it is false, the document has never been visited and an attempt is made to visit it. Thus, if the source ever does become available, it will get analyzed by the Topic Comprehension Tool and could potentially lead to the addition of more sources to the topic map.

Since the Topic Comprehension Tool retrieves sources from the Internet, the times it took to access the documents were measured. Ten trials were run, all building the topic map from scratch, starting from source 1. This gave a set of 100 access times. For the documents unable to be reached, the web servers that were contacted for those documents replied with a "HTTP 404 - File not found" error, so their response times were measured as well. The average retrieval time was 354 milliseconds, with the fastest time of 110 milliseconds and the slowest time of 3190 milliseconds. The standard deviation of the times was 561 milliseconds. The longest time it took to access all 10 documents was 7890 milliseconds (see Table 2). Since it can take up to nearly eight seconds to access 10 documents, it could take a significant amount of time to access a large number of documents.

Access times were not measured for attempts to access a non-existent web server. Such a situation would cause a significant increase in total time, due to the duration of the time-out countdown for the Resolver class of the W3C's XML parser.

8 Conclusions and Future Work

In this paper, we have described the framework for semantic-based web search as an alternative to the presently syntactic web searches available from web search engines. We have also described a tool that partially implements the proposed framework.

We are presently in the process of designing a query language based on topic metalinks and TCPs. A related extension is to investigate an efficient implementation of this language.

9 References

[And00] Anderson, Michael R., "Topic Browser: Learning From a Topic Map", MS Project, CWRU, Apr. 17, 2000.

[Auth96] "AuthorWare professional for Windows 95", *Macromedia Inc*., CA, 1996.

[AgW00] Agrawal, R., Wimmers, E.L., "A Framework for Expressing and Combining Preferences", ACM SIGMOD 2000.

[BB99] Bosworth, A., Brown Jr., A. L., "Microsoft's vision for XML", IEEE Data Eng. Bull. 22(3): 35-43, 1999.

[Bi99] Biezunski, M., "Topic Maps at a glance", at http://www.infoloom.com/tmsample/bie0.htm

[Bib1.95] Bib-1 Attribute Set Semantics, 1997, available at ftp://ftp.loc.gov/pub/z3950/defs/bib1.txt

[DC95] Weibel, S., Godby, J., Miller, E., Daniel, R., "The Dublin Core Initiative", 1995, available at http://purl.oclc.org/metadata/dublin_core

[EB00] Ozsoyoglu, G., Balkir, N.H., Cormode, G., Ozsoyoglu, Z.M., "Electronic Books in Digital Libraries", IEEE Adv. in Dig. Lib. Conf., May 2000, (Available at http://nashua.cwru.edu/TOpapers/ADL2000.pdf)

[EBa00] Ozsoyoglu, G., Balkir, N.H., Ozsoyoglu, Z.M., Cormode, G., "On the Design and Use of Electronic Books", in preparation, June 2000 (Available at http://nashua.cwru.edu/TOpapers/EBDraft.pdf)

[ECP98] Electronic Classroom Project, CWRU, 1998 (at http://erciyes.ces.cwru.edu/ecp.html).

[Epp95] Epp, Susanna S., *Discrete Mathematics with Applications*., Brooks/Cole Publishing Company. 1995.

[Gao00] Gao, Liming, "Topic Comprehension Tool for Web Users", MS Project in Progress, CWRU, 2000.

[GCMP97] Luis Gravano, Chen-Chuan K. Chang, Hector Garcia-Molina, Andreas Paepcke: STARTS: Stanford Proposal for Internet Meta-Searching (Experience Paper). SIGMOD Conference 1997: 207-218

[GILS99] Global Information Locator Service (GILS) (GILS)", 1999, available at http://www.gils.net/standards.html

[LR99] Brickley, D., Guha, R.V., "Resource Description Framework (RDF) Schema Specification" Feb. 99, at http://www.w3.org/TR/PR-rdf-schema/

[LRa99] Brickley, D., Guha, R.V., "Resource Description Framework (RDF) Schema Specification", March 1999.

[MD99] David Maier, Lois M. L. Delcambre, "Superimposed Information for the Internet", in WebDB'99 Workshop, 1999. Available at http://www.acm.org/sigmod/dblp/db/conf/webdb/webdb1999.html#LudascherPV99.

[NS99] Bray, T., Hollander, D., Layman, A., "Namespaces in XML", Jan. 1999.

[Pe99] Pepper, S., "Euler, Topic Maps, and Revolution", at http://www.infoloom.com/tmsample/pep4.htm

[Rath99] H. Rath. Technical Issues on Topic Maps. *Metastructures 99*, Montreal. Paper available online at http://www.infoloom.com/tnm/tmarticl.htm.

[Sav96] Savitch, W., "Problem Solving with C++, the Object of Programming", Addison-Wesley, 1996.

[SMIL98] Sogaj et al, "Synchronized Multimedia Integration Language (SMIL) 1.0 Specification", Standards Recommendation, World Wide Web Consortium, 1998.

[Ullm89] Ullman, J.D., "Principles of Database and Knowledge-Base Systems", Vol 1., Computer Science Press, 1989.

[WF96] Lagoze, C., Lynch, C.A., and Daniel Jr., R., "The Warwick Framework: A Container Architecture for Aggregating Sets of Metadata", available at http://cs-tr.cs.cornell.edu/Dienst/UI/2.0/Describe/ncstrl.cornell/TR96-1593

[XML98] Bray, T., Paoli, J., Sperberg-McQueen, C. M., "Extensible Markup Language 1.0 Specification". World Wide Web Consortium (W3C), February 1998.

[XMLD98] Layman, A., et al, "XML-Data", Proposal for a Standard, World Wide Web Consortium, Jan. 1998.

[XPTR99] DeRose, S., Daniel, S., Maler, E., "XML Pointer Language (XPointer)", World Wide Web Consortium (W3C), Working Draft 6 December 1999.

[XQL99] J. Robie, editor. XQL '99 Proposal, 1999, available at http://metalab.unc.edu/xql/xql-proposal.html

[XSL98] Clark, J., Deach, S., "Extensible Stylesheet Language (XSL) version 1", World Wide Web Consortium (W3C), Working Draft 16 December 1998.

[XSLT99] Clark, J, "XSL Transformations (XSLT) Version 1.0", World Wide Web Consortium (W3C), Recommendation, Nov 1999.

[Z3950] Library of Congress Maintenance Agency page for International Standard Z39.50, Jan. 2000, available at http://lcweb.loc.gov/z3950/agency