Metadata-Based Modeling of Information Resources on the Web*

S. Ayse Özel, I. Sengör Altingövde, and Özgür Ulusoy
Department of Computer Engineering, Bilkent University, 06800 Ankara, Turkey. E-mail: [selma, ismaila, oulusoy]@cs.bilkent.edu.tr

Gültekin Özsoyoğlu and Z. Meral Özsoyoğlu
Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH. E-mail: [tekin, ozsoy]@ces.cwru.edu

This paper deals with the problem of modeling Web information resources using expert knowledge and personalized user information for improved Web searching capabilities. We propose a "Web information space" model, which is composed of Web-based information resources (HTML/XML [Hypertext Markup Language/Extensible Markup Language] documents on the Web), expert advice repositories (domain-expert-specified metadata for information resources), and personalized information about users (captured as user profiles that indicate users' preferences about experts as well as users' knowledge about topics).

Expert advice, the heart of the Web information space model, is specified using topics and relationships among topics (called metalinks), along the lines of the recently proposed topic maps. Topics and metalinks constitute metadata that describe the contents of the underlying HTML/XML Web resources. The metadata specification process is semiautomated, and it exploits XML DTDs (Document Type Definition) to allow domain-expert guided mapping of DTD elements to topics and metalinks. The expert advice is stored in an object-relational database management system (DBMS).

To demonstrate the practicality and usability of the proposed Web information space model, we created a prototype expert advice repository of more than one million topics/metalinks for DBLP (Database and Logic Programming) Bibliography data set. We also present a query interface that provides sophisticated querying facilities for DBLP Bibliography resources using the expert advice repository.

Introduction

Due to the enormous growth of the World Wide Web in the last decade, today the Web hosts very large information repositories containing huge volumes of data of almost every kind of media. However, due to the lack of a centralized authority governing the Web and a strict schema characterizing the *data* on the Web—which obviously promotes this incredible growth—finding relevant information on the Web is a major struggle.

At the moment, 85% of the Internet users are reported to be using search engines for information retrieval on the Web. Most of these search engines employ either manual or automatic indexing with various refinements and optimizations (such as ranking algorithms that make use of links, etc.) (Kobayashi & Takeda, 2000). Yet, the biggest of these engines cannot cover more than 40% of the available Web pages, and the need for better search services to retrieve the most relevant information is increasing (Barfourosh, Nezhad, Anderson, & Perlis, 2002). To this end, a more recent and promising approach is indexing the Web by using metadata and annotations. It may be impossible to provide metadata for all Web resources, but still several information-rich resources and domains can benefit from such an approach. Along with the very fast approval of XML (Extensible Markup Language) (Bray, Paoli, Sperberg-McQuenn, & Maler, 2000) as a Web data exchange format, several frameworks to capture and model the Web in terms of metadata objects are proposed (i.e., Semantic Web effort [Berners-Lee, 2000] and RDF (Resource Description Framework) [1999], topic maps [Biezunski, 2001; Biezunski, Bryan, & Newcomb, 1999; Pepper, 1999], etc.).

Our goal in this paper is to exploit metadata (along the lines of the recently proposed topic maps), XML and the

* A preliminary version of this paper appears in the Proceedings of DEXA 2001. This research is supported by a joint grant from TÜBITAK (Grant No. 100U024) of Turkey and the National Science Foundation (Grant INT-9912229) of the USA.

Received July 12, 2002; revised November 15, 2002, March 13, 2003; accepted June 12, 2003

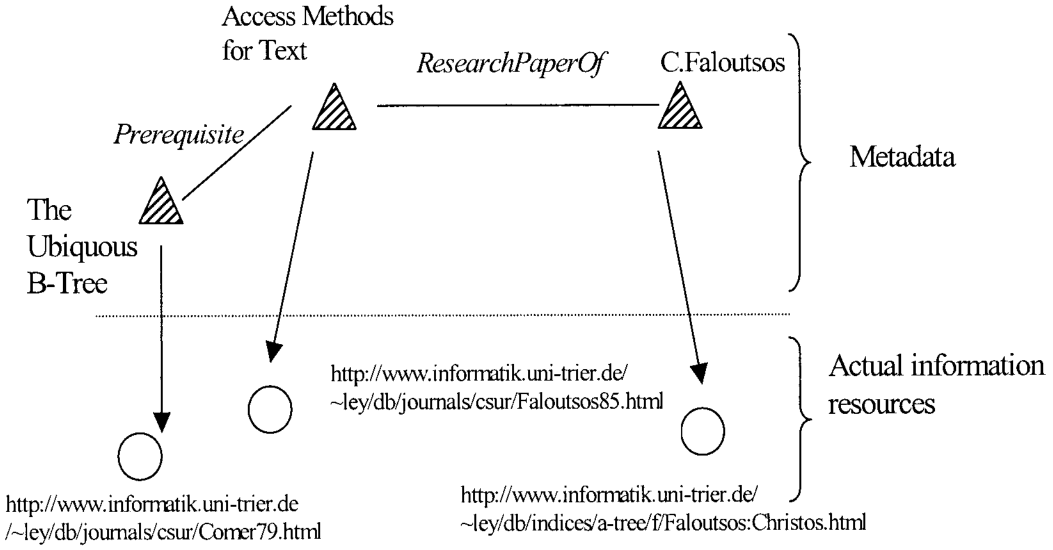

Access Methods for Text
ResearchPaperOf
C.Faloutsos
Prerequisite
The Ubiquous B-Tree
Metadata
Actual information resources
http://www.informatik.uni-trier.de/~ley/db/journals/csur/Faloutsos85.html
http://www.informatik.uni-trier.de/~ley/db/journals/csur/Comer79.html
http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/f/Faloutsos:Christos.html

FIG. 1. Metadata model for DBLP Bibliography domain defined by an expert E1.

DBMS (Database Management System) perspective to facilitate the information retrieval for arbitrarily large Web portions. We describe a "Web information space" data model for metadata-based modeling of a set of Web resources in a particular domain (i.e., *subnet*). Our data model is composed of:

- Web-based *information resources* that are XML/HTML documents.
- Independent *expert advice repositories* that contain domain expert-specified description of information resources and serve as metadata for these resources. Topics and metalinks are the fundamental components of the expert advice repositories. Topics can be anything (keyword, phrase, etc.) that characterizes the data at an underlying information resource. Metalinks are relationships among topics.
- *Personalized information* about users, captured as user profiles, that contain users' preferences as to which expert advice they would like to follow, and which to ignore, etc., and users' knowledge about the topics that they are querying.

We expect that the proposed model, used for Web querying, would lead to higher quality results compared to the results produced by a typical keyword-based searching. To illustrate the advantages of using metadata for an improved searching/querying paradigm, consider the following example.

Example 1. Assume that a researcher wants to locate all papers which are listed at the DBLP Bibliography (Ley, 2001) site, and are prerequisite papers for understanding the paper "Access methods for text" (Faloutsos, 1985) by Christos Faloutsos. Presently, such a task can be performed by extracting the titles of all papers that are cited by Faloutsos's paper and intuitively eliminating the ones which do not seem like prerequisites for understanding the original paper. Once the user manually obtains a list of papers (possibly an incomplete list), he/she retrieves each paper one by one, and examines them to see if they are really prerequisites or not. If the user desires to follow the prerequisite relationship in a recursive manner, then he/she has to repeat this process for each paper in the list iteratively. Clearly, the overall process is time-inefficient.

Instead, let us assume that an expert advice (i.e., metadata) is provided for the DBLP Bibliography site. In such a metadata model, "research paper," "Access Methods for Text," and "C. Faloutsos" would be designated as topics, and *Prerequisite* and *ResearchPaperOf* are relationships among topics (referred to *as topic metalinks*). For each topic, there would be links to Web documents containing "occurrences" of that topic (i.e., to DBLP Bibliography pages), called *topic sources*. Then, the query can be formulated over this metadata repository, which is typically stored in an object-relational DBMS, and the query result is obtained (e.g., the prerequisite paper is "The Ubiquous B-Tree" by Comer [Comer, 1979]). Figure 1 shows the metadata objects employed in this example for the DBLP Bibliography Web resources.

In Example 1, we assume that, an expert advice repository on a particular subnet (e.g., DBLP Bibliography site) is provided by a domain expert. It is also possible that different expert advice repositories may be created for the same set of Web information resource(s) to express varying viewpoints of different domain experts. Once it is formed, the expert advice repository is *stable* (i.e., changes little), stays relevant (with the exception of topic sources) even when the information resource changes over time, and is much smaller than the information resource that it models. For instance, the expert advice repository given in Example 1 captures the *ResearchPaperOf* relationship between two topics, "C. Faloutsos" and his research paper, which is a valuable and stable information even when the corresponding DBLP Bibliography resources for the paper or author are not available any more.

We make the practical assumption that the modeled information resources do not span the Web; they are defined within subnets such as the Text REtrieval Conference (TREC) series sites, or the larger domain of Microsoft Developers Network (MSDN) sites, or the very large domain of Online Collections of the Smithsonian Institution (OCSI). The creation and maintenance of metadata is an expensive process that requires vast amount of human work, if attempted in an entirely manual manner. Thus, we present an approach that exploits XML DTDs that are associated with the XML Web resources and facilitates creation of the topics and metalinks. At the moment, expert advice repositories are stored in and queried from an object-relational DBMS.

In summary, the major contributions of this paper can be listed as follows: (i) A metadata model making use of XML and topic maps paradigm is defined for Web resources; (ii) a framework to express user profiles and preferences in terms of these metadata objects is presented; (iii) a rule based approach for building metadata repositories is proposed for practically large XML subnets; and (iv) a prototype application is presented, which creates an expert advice repository for DBLP Bibliography domain to provide sophisticated querying facilities. Note that, while we essentially focus on the Web information space model in this paper, we also study issues to use this model for Web querying purposes. In Özsoyoğlu et al. (2002), we define an algebra and query processing algorithms that extend SQL (Structural Query Language) for querying expert advice repositories with some specific operators.

In the next section, we briefly discuss XML, topic maps and the related work. The "Web Information Space Model" section is devoted to the description of our Web information space model, and in the "Creation and Maintenance of Expert Advice Repositories and User Profiles" section we discuss practical issues to create and maintain expert advice repositories and user profiles. A prototype implementation is reported in the "Prototype Implementation" section. Finally, we conclude and point out future research directions in the "Conclusion" section.

Background and Related Work

Extensible Markup Language (XML) (Bray et al., 2000) is becoming a universal standard for data exchange on the Web, recommended by the W3C Consortium. XML-Data (Layman et al., 1998) 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). XML schemas specify metadata information, and allow one application on the Web to receive data from another application without any prior built-in description of the data. As an example of recent research activity on XML, see XML Special Issue (2002).

We summarize here the Topic Map data model, as described in (Biezunski, 2001; Biezunski et al., 1999; Pepper, 1999). 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 a digital scientific library for research papers (e.g., Citeseer [Giles, Bollacker & Lawrence, 1998]), each publication title or author may be a topic (e.g., "Access methods for text," "Christos Faloutsos," etc.). Topics are *typed,* (e.g., type of the topic "Christos Faloutsos" is "researcher"), and have names. Topics have *occurrences* within addressable information resources where they could be *described, mentioned*, *cited*, etc. (i.e., the topic "Access methods for text" is *described* in its corresponding pdf [portable document format] document at ACM Website). A *topic association* specifies a relationship between two or more topics. For example, the topic "Access methods for text" is *ResearchPaperOf* the topic "Christos Faloutsos" and *PublishedIn* "ACM CSUR journal". A *topic map* is a structure, perhaps a file or a database or an XML document, which contains a topic data model, together with occurrences, types, contexts, and associations. Publicly available example topic maps and topic map processors are provided in Ontopia, and Techquila company Websites. XTM (XML Topic Map) (2001) is an effort to represent topic maps as XML documents.

RDF (Resource Description Framework) (1999) is another technology for processing metadata, and it is proposed by the W3C. RDF allows descriptions of Web resources to be made available in machine understandable form. One difference of RDF from topic maps is that RDF annotates directly the information resources; topic maps, on the other hand, create a semantic network on top of the information resources. RDF is centered on resources, while topic maps on topics (Magkanaraki, Karvounarakis, Anh, Christophides, & Plexousakis, 2002). Interoperability of the two proposals is discussed in (Garshol, 2001; Lacher & Decker, 2001). Semantic Web (Berners-Lee, 2000) is an RDF schema-based effort to define an architecture for the Web, with a schema layer, logical layer, and a query language. The Semantic Web workshop (ECDL Workshop, 2000) contains various proposals and efforts for adding semantics to Web. In Magkanaraki et al., a survey on Semantic Web-related knowledge representation formalisms (i.e., RDF, topic maps, and DAML+OIL [Horrocks, 2002]) and their query languages is presented.

In WebSemantics (WS) system (Mihaila, Raschid, & Tomasic, 2002), an architecture is provided to publish and describe data sources for structured data on the World WideWeb (WWW) along with a language based on WebSQL (Mendelzon, Mihalia, & Milo, 1997) for discovering resources and querying their metadata. Our approach differs from WS in that we concentrate on the metadata model to describe and query *semistructured* (XML) Web resources that belong to a particular subnet.

The C-Web project (Christophides, 2000) is an effort to support information sharing within the specific Web communities (e.g., in Commerce, Culture, Health). The main design goals of the project include: (i) creation of conceptual models (*schema*), which could be carried out by knowl-

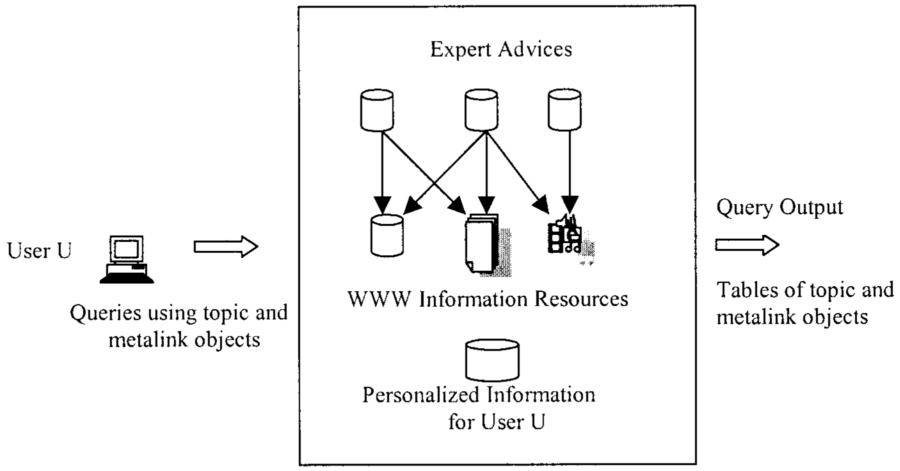

Expert Advices
User U
Queries using topic and metalink objects
WWW Information Resources
Personalized Information for User U
Query Output
Tables of topic and metalink objects

FIG. 2. Web information space model and queries.

edge engineers and domain experts and exported in RDF syntax, (ii) publishing information resources using the terminology of conceptual schema, and (iii) enabling community members to query and retrieve the published information resources. The querying facilities are provided by the language, so-called, RQL. Note that, the basic ideas and motivation of C-Web project and our work are quite similar, but the approaches for modeling, storing, and querying the metadata differ.

In this paper, we essentially describe the Web information space model with expert advice repositories at its heart, and focus on the properties of topic/metalink based model, as well as some practical issues for their automated creation. In Özsoyoğlu et al. (2002), we also describe SQL extensions for querying expert advice repositories for Web querying purposes, along with an algebra and query processing algorithms. While we propose a metadata-based search and querying paradigm for specific subnets on the Web, the previously proposed Web query languages in the literature (see Florescu, Levy, & Mendelzon [1998] for an extensive survey) have the broader goal of querying the Web as a whole.

Web Information Space Model

In this section, we present our Web information space model, which is illustrated in Figure 2. The three components of the model are information resources, the expert advice model, and the user profile model.

Information Resources

Information resources are Web-based documents, containing multimedia data of any arbitrary type. They may have bulk text in various formats (e.g., ascii, PostScript, pdf, etc.), images with different formats (e.g., jpeg), audio, video, audio/video, etc. For the purposes of this research, we assume that information resources are in the form of XML/HTML documents.

Topic source is an information resource in which a particular topic occurs. For example, the pdf document for the topic (with name) "Access methods for text" and all other documents that cite this topic in ACM Portal Website constitute a topic source for this topic. For XML-based Web documents, we assume that a number of topic source attributes are defined within the XML document (using XML element tags) such as *LastUpdated*, *Author*, and *MediaType* attributes, etc. Also, the expert advice model, discussed next, has an entity, called Topic Source Reference, which contains (partial) information about a topic source (such as its Web address, etc.).

Expert Advice Model

In the proposed Web information space model, expert advices are metadata that describe the contents of associated information resources. Each domain expert models a *subnet* (a set of information resources in a particular domain) in terms of:

i. Topic and topic source reference entities
ii. Metalinks (i.e., metalink types, signatures and instances).

Expert advice repositories are stored in a traditional object-relational DBMS. In particular, there is a table for topics, topic source references, and each metalink type (see Table 1). We assume that expert advice repositories are made available by the associated institutions (e.g., DBLP Bibliography Website) to be used for sophisticated querying purposes. Besides, independent domain experts (i.e., so-called *information brokers* (Rath & Pepper, 1999)) could also publish expert advice repositories for particular subnets on their Websites as a (probably feed) service. Please note that a semiautomated means of creating such repositories is discussed in the next main section, after we describe the properties of the model in detail.

TABLE 1. Topics and metalinks tables.

(a) Topics table

Tid	TName	TType	TDomain	TImp
T01	Christos Faloutsos	Researcher	Information retrieval	. . .
T08	Access methods for text	Research paper	Information retrieval	. . .

(b) ResearchPaperOf metalink table

Mid	ResearcherId	PaperId	MImp
M01	T01	T08	. . .

Topic and Topic Source Reference Entity Types. We start with the topic entity, which has the following attributes:

- *T(opic-)Name* (of type string) contains either a single word (i.e., a keyword) or multiple words (i.e., a phrase). Topic names characterize the data (real-world subjects [XML Topic Maps, 2001]) in information resources. Example topic names are "database" (a keyword), "text compression techniques" (a phrase), and "Access methods for text" (the title of a paper by Faloutsos [Faloutsos, 1985]). Topic names are defined by domain experts, and can be arbitrarily specified phrases or words.
- *T(opic-)Type* and *T(opic-)Domain* attributes specify, respectively, the type of the topic and the domain within which the topic is to be used. For example, the topic "Access methods for text" is of type "research paper" in the domain of "information retrieval." Again, we allow different experts to use different words/phrases for topic types and topic domains.
- *T(opic-)Author* attribute defines the expert (name or id or simply a URL [Universal Resource Locater] that uniquely identifies the expert) who authors the topic.
- *T(opic-)MaxDetailLevel*. Each topic can be represented by a topic source in the Web information resource at a different *detail level*. Therefore, each topic entity has a maximum detail level attribute. Let us assume that levels 1, 2, and 3 denote levels "beginner," "intermediate," and "advanced." For the "information retrieval" domain, for example, a source for topic "indexing" can be at a beginner (i.e., detail level 1) level, denoted by Indexing^1 (e.g., "Inverted File Indexing"). Or it may be at an advanced (say, detail level *n*) level of Indexing^n (e.g., "Compressed In-Memory Inversion"). Note the convention that topic x at detail level i is more advanced (i.e., more detailed) than topic x at detail level j when $i > j$.
- *T(opic-)id*. Each topic entity has a T(opic-)id attribute, whose value is an artificially generated identifier, internally used for efficient implementation purposes, and not available to users.
- *T(opic-)SourceRef*. Each topic entity has a T(opic-)SourceRef attribute which contains a set of Topic-Source-Reference entities as discussed below.
- *T(opic)-Importance-Score*. Each topic entity has a T(opic)-Importance-Score attribute whose value represents the "importance" of the topic. Experts assign importance scores to topics in manual/semiautomated/automated manner, which is discussed in the "Attaching Importance Scores to Metadata Entities" section.
- Topics also have other attributes such as roles, role-playing, etc. Some of these additional attributes are discussed in the topic map standard and described in detail in XML Topic Maps (2001).

The attributes (TName, TType, TDomain, TAuthor) constitute a key for the topic entity, and the Tid attribute is also a key for topics.

A *T(opic-)S(ource-)Ref(erence)*, also an entity in the expert advice model, contains additional information about topic sources. A topic source reference entity has the following attributes.

- *Topics* (set of Tid values) attribute that represents the set of topics for which the referenced source is a topic source.
- *Web-Address* (URL) of the topic source.
- *Start-Marker* (address) indicating the exact starting address of the topic source relative to the beginning of the information resources (e.g., http://informationRetrieval.org/indexing#invertedFile).
- *Detail level* (sequence of integers). Each topic source reference has a *detail level* describing how advanced the level of the topic source is for the corresponding topic.
- Other attributes such as *S(ource)-Importance-Score*, *Media-type*, *Role*, *Last-Modified*, etc.

Metalink Entities. Topic Metalinks represent relationships among topics. Metalinks have attributes such as type, domain, and importance-score, as described for topic entities. A simple metalink type is → *ResearchPaperOf.* The notation (*ResearchPaperOf* represents an instance of this metalink type, as in "Access methods for text" (*ResearchPaperOf* "C. Faloutsos," and this metalink instance states that "Access methods for text" is a research paper of "C. Faloutsos."

In expert advice repositories, domain experts specify both the metalink signatures and metalink instances. A *metalink signature* serves as a definition for a particular metalink type, and includes the name given to the metalink type and the topic types of topics that are related with this metalink type. For instance, the signature "*ResearchPaperOf*(E): research paper → (SetOf (researcher)" denotes that the *ResearchPaperOf* metalink type can hold between topics of types "researcher" and "research paper."

Another metalink type is *Prerequisite* given with the signature *Prerequisite*(E): SetOf (topic) → (SetOf (topic). The metalink instance "Inverted Index Structures"^{2} → (*Prerequisite* "Text Indexing" ^{1} states that "Understanding of the topic "inverted index structures" at level 2 (or higher) is the prerequisite for understanding the topic "text indexing" at level 1." Yet another metalink relationship can be the *RelatedTo* relationship that states, for example, that the topic "relational model" is *RelatedTo* the topic "normalization theory." *SubTopicOf* and *SuperTopicOf* metalink types together represent a topic composition hierarchy. As an example, the topic "database" is a supertopic (composed) of topics "data model," "query languages," "query processing," etc. And the topic "relational algebra" is a subtopic of "query languages" and "relational model."

Thus any relationship involving topics deemed suitable by an expert in the field can be a topic metalink. Metalinks represent relationships among topics, not topic sources. Therefore, they are "meta" relationships, hence our choice of the term "metalink."

Topic Closure. Metalink types, representing relationships, can have several properties such as transitivity, reflexivity, etc. For example, *RelatedTo* is both transitive and reflexive. *SubTopicOf* and *Prerequisite* are transitive, but not reflexive. For the metalink types with transitivity, we may like to follow metalink instances to obtain a complete set of topics that are related through this particular metalink type. For instance, assume that the expert E specifies three topic entities A (with name "Relational Algebra"), B ("SQL"), and C ("RDBMS application"), and the metalink instances $A \rightarrow$ *Prerequisite* B, and $B \rightarrow$ *Prerequisite* C. (For the sake of simplicity, assume that all detail level values are 1, and ignored.) And, the user U asks for all sources for topic C and its prerequisites, followed recursively, subject to the advice of expert E. Since the *Prerequisite* metalink is transitive, topic sources for A and B are also included in the result of this request.

In general, we define the notion of *topic closure* of a set of topics X with respect to a metalink type M as all topics that are reached by following the metalink instances with that particular metalink type. Intuitively, topic closure computation algorithm starts expanding the initial set X with topics Z such that $Z \rightarrow M\ Y$, where $Y \subseteq X$, and continues in the same manner recursively. Topics and metalink instances can be represented as a graph in which topics and metalinks correspond to nodes and labeled edges, respectively. Then, a graph search algorithm (like BFS [Breadth First Search]) can be used to compute topic closure of a set of topics with respect to a metalink type.

Note that metalink signatures allow either side of metalink instances to be sets of topics. For instance, assume that an expert declares a topic entity Z ("QUEL") and a metalink instance (B, Z) (*Prerequisite* C, which means that both SQL and QUEL (Query Language) are prerequisites for RDBMS (Relational Database Management System) application. In this case, we need to know whether we can decompose the left-hand side of a *Prerequisite* metalink to compute the closure correctly. That is, if (B, Z) $\rightarrow$ *Prerequisite* C is equivalent to $B \rightarrow$ *Prerequisite* C, and $Z \rightarrow$ *Prerequisite* C, then the prerequisites for topic C include B, Z, and A. If the equivalence does not hold, the prerequisites are B and Z, but not A. Clearly, for the *Prerequisite* metalink, the former option is more intuitive. So, besides declaring the metalink signatures and instances, the domain expert should also specify properties such as the reflexivity, transitivity, decomposability of the metalinks. Then, user queries that require computing the closure of a topic with respect to a (set of) metalink(s) can be safely evaluated. In Özsoyoğlu et al. (2002), we specify topic closure computation algorithms for more general cases, i.e., regular expressions of metalink types.

Attaching Importance Scores to Metadata Entities. Besides describing information resources through topics and their metalinks, a domain expert further attaches importance scores to these descriptive metadata entities for providing more sophisticated querying facilities. Adding importance scores to topics, their sources, and metalinks enriches the Web information space model by allowing query output ranking and size control. A query output is ranked with respect to metadata importance scores and limited to the highest-ranked topics/sources to save query processing time and improve the quality of query results. Our model enables the user to formulate queries[1] such as the one given in the example below.

Example 2. Find the top five most important topics and their sources that are prerequisites for understanding the topic "Relational Algebra."

An importance score is a real number in the range [0, 1], and it can also take its value from the set {No, Don't-Care} . The importance score is a measure for the importance of the topic, except for the cases below.

a. When the importance value is "No," for the expert, the metadata object is rejected (which is different from the importance value of zero in which case the object is accepted, and the expert attaches a zero value to it). In other words, metadata objects with importance score "No" are not returned to users as query output.
b. When the importance value is "Don't-Care," the expert does not care about the use of the metadata object (but will not object if other experts use it), and chooses not to attach any value to it. Please see Example 3 for more detail.

Importance scores are attached to metadata entities in different forms, namely,

a. *Open form* (Agrawal & Wimmers, 2000): For each metadata object in the repository, an importance score is specified. As an example, we may have Imp(E.Topics, TName = "Access methods for text," TType = "research paper," TDomain = "IR") = 0.9, where Imp( ) denotes (a constant) importance score function, E.Topics denotes the topics table of the expert advice repository created by the expert E, and IR denotes information retrieval. This statement expresses that the domain expert assigns the importance score of 0.9 to the topic (paper) "Access methods for text" in the "IR" domain. Note that, in the open form, the domain expert manually assigns importance scores to metadata objects, which is an unlikely situation except for very small expert advice repositories.

[1] The syntax of our query language extensions is defined elsewhere (Özsoyoğlu et al., 2002).

b. *Closed form*: Each object's importance score is derived from a closed function. This approach is more practical to apply during automated or semiautomated metadata creation. For instance, the importance score for topics of type "research paper" can be specified as a weighted function of citations received and the impact factor of the journal in which the paper is published. Then, we express the importance score function in the closed form as Imp(E.Topics, TType = "research paper") = f(no of citations, impact factor of the journal). In this case, the domain expert should also specify how to compute the function *f*( ) and determine each parameter in this function, i.e., the number of citations received by the paper and impact factor of the journal that the paper is published in. Clearly, such importance score functions are usually domain (or application) dependent and should be specified by the domain expert.

c. *Semiclosed form*: A function specifies a score for a set of objects identified through regular expressions. Consider the function Imp(E.TopicSources, TName="*processor speed*," TDomain = "computer hardware," Last-Modified = (Now − 2years)) = No, where * denotes a wildcard character that matches any string. This function assigns the importance score "No" to all Web resources for any topic with topic name including the string "processor speed" in the "computer hardware" domain and not updated in the last 2 years. So, these topic sources will never be included in query outputs unless they are updated. The function Imp(E.TopicSources, TDomain = "information retrieval," Web-Address = "http://trec.nist.gov") = 1 implies that the domain expert considers all resources at the TREC site as extremely important for topics in the information retrieval domain. As other examples, consider:

Imp(E.Prerequisite, Relational Algebra^1 → (*Prerequisite* SQL^1) = 0.9,
Imp(E.Prerequisite, Circuit Design → (*Prerequisite* SQL) = No,
Imp(E.Prerequisite, QUEL → (*Prerequisite* SQL) = Don't- care.

The first score assignment states that the importance score of the metalink instance "the topic "Relational Algebra" at the beginner level (1) is prerequisite to "SQL" at the beginner level" is very high (0.9). Note that each metalink type is stored in its corresponding table in the expert advice repository, e.g., E.Prerequisite captures all instances of this particular metalink type. The second score assignment states that understanding "circuit design" is not a prerequisite for understanding the topic "SQL." In the last statement, the expert E does not consider the topic "QUEL" as a prerequisite to understand topic "SQL," but also does not object to those experts that do think so.

Personalized Information Model: User Profiles

The user profile model maintains for each user his/her preferences about experts, topics, sources, and metalinks as well as the user's knowledge about topics.

User Preferences. In this paper, we employ user preference specifications, along the lines of Agrawal and Wimmers (2000). The user U specifies his/her preferences as a list of Accept-Expert, T(opic)-Importance, etc. statements, as shown in Example 3. Essentially, these preferences indicate in which manners the expert advice repositories can be employed while querying underlying information resources. In this sense, they may affect the query processing strategies for, say, a query language or a higher-level application that operates on the Web information space model.

In particular, the Accept-Expert statement captures the list of expert advice repositories (their URLs) that a user relies and would like to use for querying. Next, T(opic)-Importance and S(ource)-Importance statements allow users to specify a threshold value to indicate that only topics, or topic sources with greater importance scores than this threshold value are going to be used during query processing and included in the query outputs. Furthermore, the users can express (through Reject-T and Reject-S statements) that they do not want a topic with particular name, type, etc., or a topic source at a certain location to be included in the query outputs, regardless of their importance scores. Finally, when there are more than one expert advice repositories it is possible that different experts assign different importance scores to the same metadata entities. In this case, the score assignments are accepted in an ordered manner as listed by the Accept-Expert statement. We illustrate user preferences with an example.

Example 3. Assume that we have three experts www.distributed-cs.org (E1), www.networkcomputing.org (E2), and www.ai-resources.org (E3). The user John-Doe is a researcher on distributed systems and specifies the following preferences:

Accept-Expert (John-Doe) = {E1, E2},
T-Importance (John-Doe) = {(E1, 0.9), (E2, 0.5)},
S-Importance (John-Doe) = {(E1, 0.5)},
Reject-T (John-Doe) = {(E2, TName= "*parallel*")},
Reject-S (John-Doe)={Web-Address= www.hackersalliance.org}.

Note that the user preferences are practically stored in an object-relational DBMS; in this example; preferences are shown as a list of statements for the sake of comprehensibility. The first preference states that Professor Doe wants to use expert advice repositories E1 and E2 to query the underlying Web resources, but not E3 (which includes metadata about irrelevant resources to his research area). The second and third clauses further constrain that only topics and sources with importance values greater than the specified threshold values should be returned as query output. For instance, a topic from repository E1 will be retrieved only if its importance score is greater than 0.9. The fourth preference expresses that Professor Doe does not want to see any topics that include the term "parallel" in its name from the repository E2, as he is only interested in distributed systems' issues. The fifth one forbids any resource from the (imaginary) site www.hackersalliance.org to be included in any query outputs. Finally, if there is a conflict in the importance score assigned to a particular topic or source by experts E1 and E2, then, first, advices of E1 and then only nonconflicting advices from E2 are accepted. For example, assume that topic "name transparency" has importance score 0.9 in E1 and "No" in E2, then the topic "name transparency" is included in the query results, since the conflicting advice from E2 is not considered. As another example, assume that expert E1 assigns importance score of "Don't Care" for topic "distributed query processing" and expert E2 assigns 0.6 importance score for that topic. Then, the topic is included in the query results, given that E1 does not care whether the topic is included or not, but E2 assigns importance score of 0.6, which is greater than the threshold value specified in the T-Importance statement.

User Knowledge. For a given user and a topic, the knowledge level of the user on the topic is a certain detail level of that topic. The knowledge level on a topic cannot exceed the maximum detail level of the topic. The set U-Knowledge (U) = {(topic, detail-level-value)} contains users' knowledge on topics in terms of detail levels. As in other specifications, topics may be fully defined using the three key attributes TName, TType, and TDomain, or they may be partially specified in which case the user's knowledge spans a set of topics satisfying the given attributes. We give an example.

Example 4. Assume that the user John-Doe knows topics with names "distributed query processing" at an expert (3) level, and "distributed transaction management" at a beginner (1) level, specified as UKnowledge (John-Doe) = {(TName = "distributed query processing", 3), (TName = "distributed transaction management", 1)}.

Besides detail levels, we also keep the following history information for each topic source that the user has visited: Web addresses (URLs) of topic sources, their first/last visit dates, and number of times the source is visited. The information on user's knowledge can be used while evaluating query conditions and computing topic closures, in order to reduce the size of the information returned to the user (Altingövde, Ozel, Ulusoy, Özsoyoğlu, & Özsoyoğlu, 2001a,b). 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.

Creation and Maintenance of Expert Advice Repositories and User Profiles

In this section, we briefly discuss some issues to demonstrate that the proposed Web information space model is practically applicable. In particular, we focus on the following questions: (i) How the expert advice repositories are created and maintained and how scalable they are, (ii) how the metadata objects from various expert advice repositories (as well as the user queries) are matched, and (iii) how the user profiles are constructed. While answering these questions, we either propose our own solutions or present the ways in which current technologies and approaches may be adapted to the problem at hand.

Automated Creation and Maintenance of Metadata Objects

The approach proposed in our work does not address the whole *Web* to solve the problem of information retrieval (IR), but we rather concentrate on the so-called *subnets* for which the creation and maintenance of metadata is an attainable task. However, the description and size of such subnets may be both large and diversified enough to be still able to benefit from the model proposed here. As exemplified in the previous section, a collection of Web sites that belong to a particular domain (such as DBLP Bibliography) can be the target subnet. Furthermore, it is not necessary that the set of information resources associated with a particular metadata repository should be physically in the same domain, or even belong to the same establishment. For instance, we may like to create a semantic index for all computer science papers (i.e., similar to the Citeseer [Giles, Bollacker, & Lawrence, 1998] search engine) where the indexed resources can be found anywhere on the Web. In particular, the ideas we discuss in this section are similar to those described in the C-Web project (Christophides, 2000).

As an initial step, the topic and metalink types are determined for the application domain. This is carried on by the domain experts either in a totally manual manner or by making use of thesauri or available ontologies (if any). The more crucial step is extracting metadata from the actual Web resources and this may involve techniques from the subfields like machine learning, data mining and/or information retrieval (see Folch & Habert [2000] and Witten, Paynter, Frank, Gutwin, & Nevill-Manning [1999] as examples). We envision that, the advent of the XML over the Web can further facilitate such automated processes and allow constructing tools that will accurately and efficiently gather metadata for arbitrarily large subnets, with least possible human intervention. Essentially, our strategy is mapping topics and metalinks to the elements and attributes of XML DTDs. Given a set of DTDs, domain experts designate the values of a number of elements (or attributes) as topics, and further define particular relationships (metalinks) among these topics. Then, a *Web robot*-similar to those employed in today's search engines or focused crawlers- traverses the Web, creates metadata objects with respect to the mapping of the domain expert for each document conformant to the given DTDs, and stores the metadata entities in a DBMS.

To illustrate this semiautomated approach, consider the (simplified) DTD given in Figure 3 for the DBLP Bibliography archive. Assume that a domain expert defines the mapping *M* in Figure 4 for DBLP DTD. In this mapping, the first line specifies that the following element and attribute

```
<!ELEMENT dblp (article|
inproceedings|proceedings|
book|...)*>
<!ENTITY % field "author|editor|
title|booktitle|year|
address|journal|URL">

<!ELEMENT article        (%field;)*>
<!ELEMENT inproceedings  (%field;)*>
<!ELEMENT proceedings    (%field;)*>

<!ELEMENT author     (#PCDATA)>
<!ELEMENT editor     (#PCDATA)>
<!ELEMENT address    (#PCDATA)>
...
```

FIG. 3. DBLP DTD (simplified).

names would be in the namespaces of the DBLP DTD. Next, *generic topics* are defined in such a way that each distinct paper title or author contributes to a topic entity. For instance, the specification of *t1* forces the Web-robot to create topics whose TName values are extracted from the path *dblp.article.title* in any XML document conformant to the DTD. The TType of the topic is designated as "research paper."

Next, metalink generation rules are defined. For instance, *ResearchPaperOf* metalink specifies the relationship between a paper and its author(s). The parentOf predicate states that both the author and title should belong to the same XML subtree to participate in a particular metalink instance. Note that the specification of metalinks can involve far more complex constraints/functions that may be determined according to the application or domain requirements (please see the "Prototype Implementation" section). Finally, the source generation rule states that the URL value obtained from the path *dblp.article.URL* is the Web address of the topic source for topic *t1*.

```
M = { nms1: "http://dblp.org/dblp.dtd",
// topic generation rules
t1: <nms1, TName = ValueOf (dblp.article.title),
            TType = research paper>,
t2:<nms1, TName = ValueOf (dblp.article.author),
                TType = researcher>,

// metalink generation rules
m1: <ResearchPaperOf: t1→ t2 |
     parentOf(t1) = parentOf(t2)>

// source generation rules
s1: < Web-Address = ValueOf (dblp.article.URL),
      Topics = {t1}> }
```

FIG. 4. Mapping *M* for metadata extraction.

```
<?xml version="1.0"?>
<!DOCTYPE dblp SYSTEM "dblp.dtd">
<dblp>
  <article key="...">
    <title> Access Methods for Text </title>
    <author>Christos Faloutsos</author>
    <pages>...</pages>
    <crossref>...</crossref>
    <year>1985</year>
    <journal>ACM Computing Surveys</journal>
    <url>http://www.informatik.uni-trier.de/
       ~ley/db/journals/csur/Faloutsos85.html
    </url>
  </article>
</dblp>
```

FIG. 5. Example XML document.

A domain expert may provide such a mapping simply by interacting with a GUI-based tool, or by specifying a set of rules, or in any other convenient manner. Given such a mapping, a Web-robot traverses the Web and creates the expert advice database entries for any document that conforms to the DTD. Let us assume that the Web-robot encounters the example XML document of Figure 5 at DBLP Bibliography Web site. Then, the metadata objects similar to those given in Table 1 will be created. Note that the URL extracted from this document serves as a topic source for the topic "Access Methods for Text" and the XML document itself from which the metadata is extracted is another topic source for both topics. Topic sources are again stored in a database table, which is not shown due to space considerations.

Once we store the metadata as shown in the Table 1, sophisticated queries can be easily posed for the underlying Web resources. Thus, the metadata repository covers a distributed set of documents over the Web, while providing the querying power of a central database system. For instance, using an expert advice repository created as in the above manner (with *ResearchPaperOf* and *Prerequisite* metalink types), one can pose the query "find all research papers which are written by Christos Faloutsos and prerequisite of some paper *P*," in an efficient manner with highly qualified results, whereas it could be quite difficult to obtain the same result using traditional search engines over the Web.

Matching Metadata Objects from Multiple Expert Advices

As mentioned before, different expert advice repositories may be created for the same set of Web information resource(s) to express varying viewpoints of different domain experts. If a user desires to make use of more than one expert advice repositories for querying the underlying subnet, the issue of comparing and combining metadata objects from different repositories arises. A primary difficulty for querying a set of expert advice repositories simultaneously is inferring whether two (or more) experts are talking about the same object or not. For instance, one domain expert may associate a particular Web page with topic "inverted index" whereas another one names the associated topic as "inverted file." An application using these repositories should infer that these two experts refer to the same topic, and respond to the query accordingly. Furthermore, suppose that an end-user asks for all the topics that are prerequisite to understand "inverted index structures." Now, the application should discover that all three topics are about the same concept. To solve this problem of matching metadata objects specified by different experts or mentioned by end-users, we rely on the following mechanisms:

- *Subject-based matching*: Intuitively, the idea is using a Web resource to serve as a certain definition, which is publicly agreed upon, for a particular topic. In topic maps standard, such reference resources are called public subject indicators (PSI). For instance, the topic "XML" is formally defined in a W3C specification (Bray et al., 2000) with a particular URL, which can actually serve as a correct, unambiguous and universal definition for the XML for all experts who want to use this topic while describing information resources. So, in two different expert advice repositories there may be two different topics with names, say "XML" and "Extensible Markup Language," which point to the same Web address as their public subject indicator. Then, an application processing these two repositories can automatically understand that these two topics are indeed about the same concept. At the moment, Web accessible thesauri or ontology (such as WordNet [1998]) can serve as such PSI directories.
- *Name-similarity-based matching*: In this case, the name similarity between the metadata objects is used to decide whether two metalink objects refer to the same concept. For instance, two topics with the same type in the same domain may be matched if their names are similar above a prespecified threshold value. The similarity of phrases may be decided using one of the various well-known techniques, such as edit-distance measures or vector space models (Salton, 1989), whichever is most appropriate for the application domain. For instance, let us consider two topics specified in different repositories E1 and E2 with topic names "inverted index" and "inverted index structure." The name similarity of these topics can be computed by using the vector space model by an application that accesses both repositories. If the similarity degree is, say, found to be 90%, then the application decides that the topics are the same.

Please note that, among two approaches we describe above, the former one is more suitable for automatic inference of metadata equivalence in different expert advice repositories, whereas the latter is more practical to match metadata entities specified in user queries with those entities specified in the expert advice repositories.

Creation and Maintenance of User Profiles

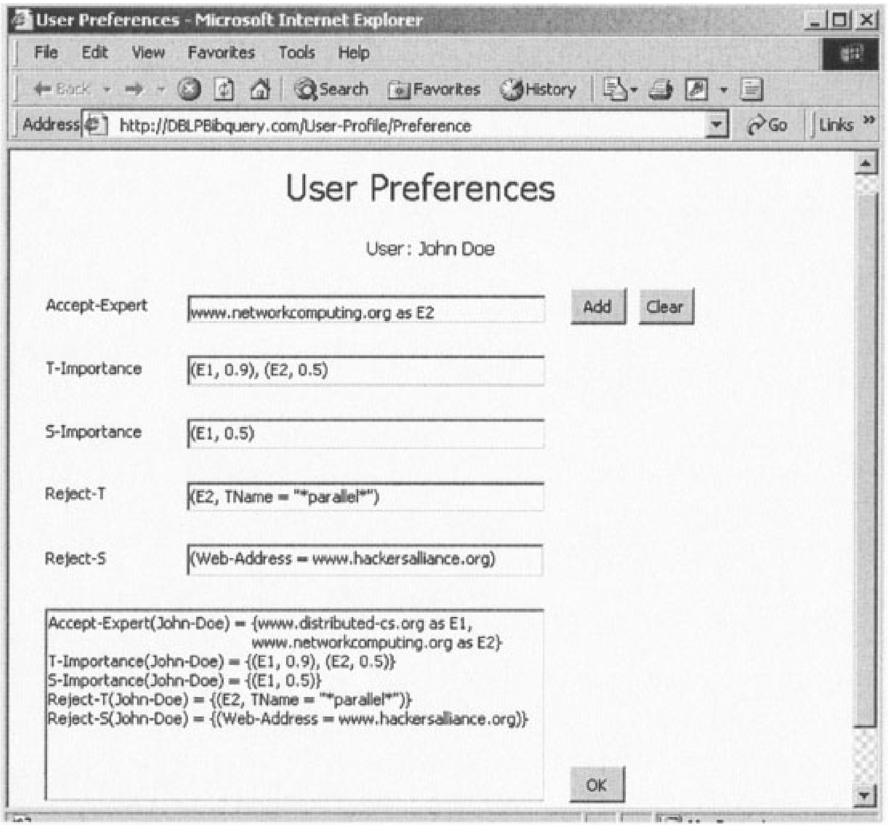
In our Web information space model, user profiles are composed of user preferences and user knowledge. User preferences allow each user to specify his/her preferences about experts and metadata objects. User knowledge maintains knowledge of users on topics in terms of detail levels as well as navigational history information for the users. We create and store our Web information space user profile in the server side, and each user should first login to the system and then he/she can create his/her preferences by filling out a form, as shown in Figure 6 (i.e., user-created profile [Kuflik & Shoval, 2000]). For the time being, the user should explicitly specify which expert advice repositories he/she wants to use, as well as the other preferences (topic importance threshold, rejected resources, etc.). User knowledge can be generated and updated from user click-stream data that is collected at the application level, i.e., search/query interface for a Web querying application based on our model. Assume that a user who login to such a Web querying application poses a query involving various metadata entities and a list of required topic source URLs is returned. Then, as the user clicks some links in this list, the URL of the document that the user visits, the first and last visit dates, media type, and the visit frequency for the document are directly written to the user knowledge database. Besides, the detail level of each such topic source for the required topic in query is retrieved from the expert advice repository and stored in the user knowledge.

Example 5. Assume that the user John Doe requires all sources for the topic "inverted index," and the expert advice includes three sources S1, S2, S3 for this topic with detail levels beginner (1), intermediate (2), and advanced (3), respectively. All three sources are returned to the user as the query response. Assume that the user knowledge formerly includes the entry UKnowledge (John-Doe) = {(TName = "inverted index", 1)} and the user clicks to S1 and S2. Then, his knowledge about this topic will be updated as "intermediate" and the entry becomes UKnowledge (John-Doe) = {(TName = "inverted index", 2)}. Moreover, the list of visited URLs by the user John Doe is expanded with sources S1 and S2, along with their visit dates, media types, etc. The user knowledge database for this example is shown in Table 2.

Information captured in user profile is employed for refining query results that are initially obtained by querying expert advice repositories. For instance, the user John Doe would specify that the sources for the topic "inverted index" should be eliminated from the query output if he has visited these resources in the last two weeks or the sources are at the "beginner" level. We discuss elsewhere the use of user profiles for query output refinement purposes in more detail (Altingövde, Özel, Ulusoy, Özsoyoğlu, & Özsoyoğlu, 2001a,b).

Prototype Implementation

DBLP Bibliography is a Web service with bibliographic information on computer science publications in major journals and conference proceedings (Ley, 2001). In this section, we outline a prototype application that creates an expert advice repository for DBLP Bibliography Web site and queries this repository. In particular, DBLP Bibliogra-


User Preferences - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
History
Address
Go
Links
Add
Clear
OK

Address: http://DBLPBibquery.com/User-Profile/Preference

User Preferences

User : John Doe

Accept-Expert	www.networkcomputing.org as E2
T-Importance	(E1, 0.9), (E2, 0.5)
S-Importance	(E1, 0.5)
Reject-T	(E2, TName = "*parallel*")
Reject-S	(Web-Address = www.hackersalliance.org)

```
Accept-Expert(John-Doe) = {www.distributed-cs.org as E1,
                           www.networkcomputing.org as E2}
T-Importance(John-Doe) = {(E1, 0.9), (E2, 0.5)}
S-Importance(John-Doe) = {(E1, 0.5)}
Reject-T(John-Doe) = {(E2, TName = "*parallel*")}
Reject-S(John-Doe) = {(Web-Address = www.hackersalliance.org)}
```

FIG. 6. User preference specification form.

phy exports its contents in XML format including entries for more than 200,000 publications (approximately 90 MB in size). The XML files conform to the DBLP DTD, which is given in the "Automated Creation and Maintenance of Metadata Objects" section in a simplified manner. At the moment, all of the XML files that conform to this DTD are those files at the DBLP Bibliography site, so the prototype application does not crawl the Web to reach them. In the future, we expect that DBLP DTD can be used by some paper authors, who want to be indexed by the DBLP Website (or a research paper repository, as we outline here), given the existence of a robot that would crawl the Web and gather the documents conformant to the DTD. For instance, Professor John Doe puts DBLP-conformant XML files for his publications at his Web site, to be indexed by any paper repository that recognizes DBLP DTD.

TABLE 2. User knowledge tables.

(a) Navigational history information for user John Doe

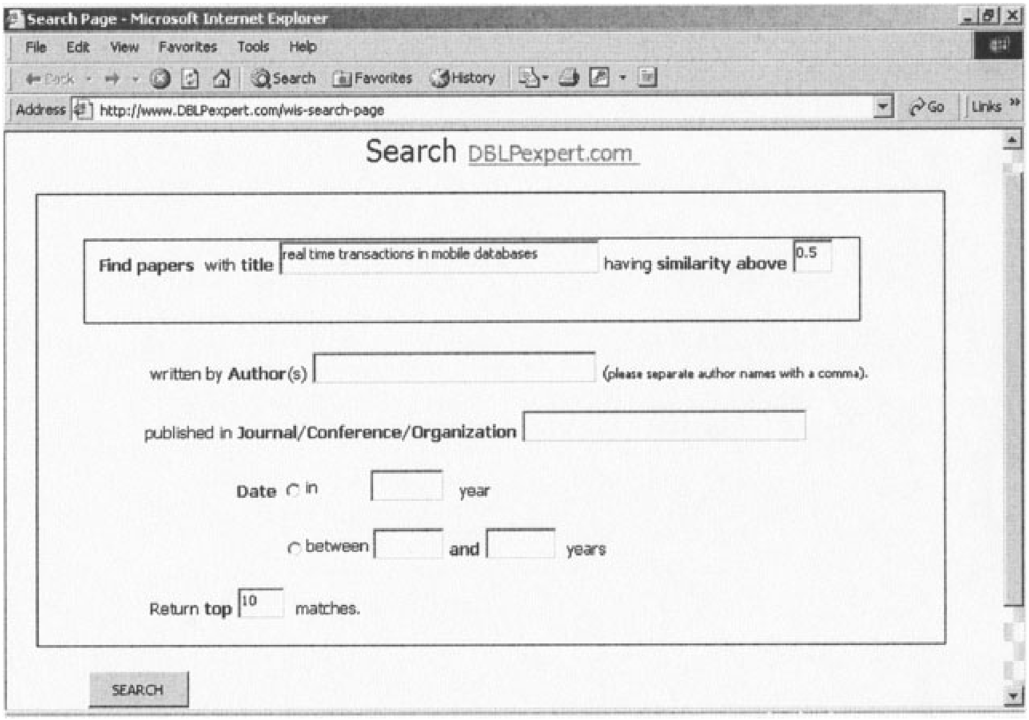
TName	Detail Level	URL	Source Role	Source MType	First Visit	Last Visit	Freq
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
Inverted index	1	www . . .	. . .	. . .	1.2.02	2.3.03	11
Inverted index	2	www . . .	. . .	. . .	2.3.03	2.3.03	1
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .

(b) Topic knowledge for user John Doe

Tname	TType	TDomain	TAuthor	Knowledge Level
Inverted index	. . .	. . .	. . .	2
. . .	. . .	. . .	. . .	. . .

Based on DBLP DTD, we define the following metadata entities whose instances are automatically extracted from the XML files as described in the "Automated Creation and Maintenance of Metadata Objects" section:

- *Topic types*: ResearchPaper, Researcher, JournalConfOrganization, PublicationDate, and PublicationYear. We specify a mapping between each of these topic types and DTD paths. For instance, the topic name for a topic of type "ResearchPaper" is extracted from the paths *dblp.article.title* or *dblp.proceedings.title*, etc. Since the mappings are straightforward, we do not specify them here explicitly.
- *Metalink types*: For this application domain, we define the metalink types *ResearchPaperOf*, *PublicationDate*, *PublishedInOrg*, *RelatedTo*, and *Prerequisite*. The first three metalink types are easily defined according to the corresponding paths in the DTD, and not discussed here. The latter two are more complex. *RelatedTo* metalink type expresses a relationship between two topics of type "ResearchPaper."


Search Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://www.DBLPexpert.com/wis-search-page
Go Links »
SEARCH

Search DBLPexpert.com

Find papers with title real time transactions in mobile databases having similarity above 0.5

written by Author(s) (please separate author names with a comma).

published in Journal/Conference/Organization

Date ○ in year

○ between and years

Return top 10 matches.

FIG. 7. Search page.

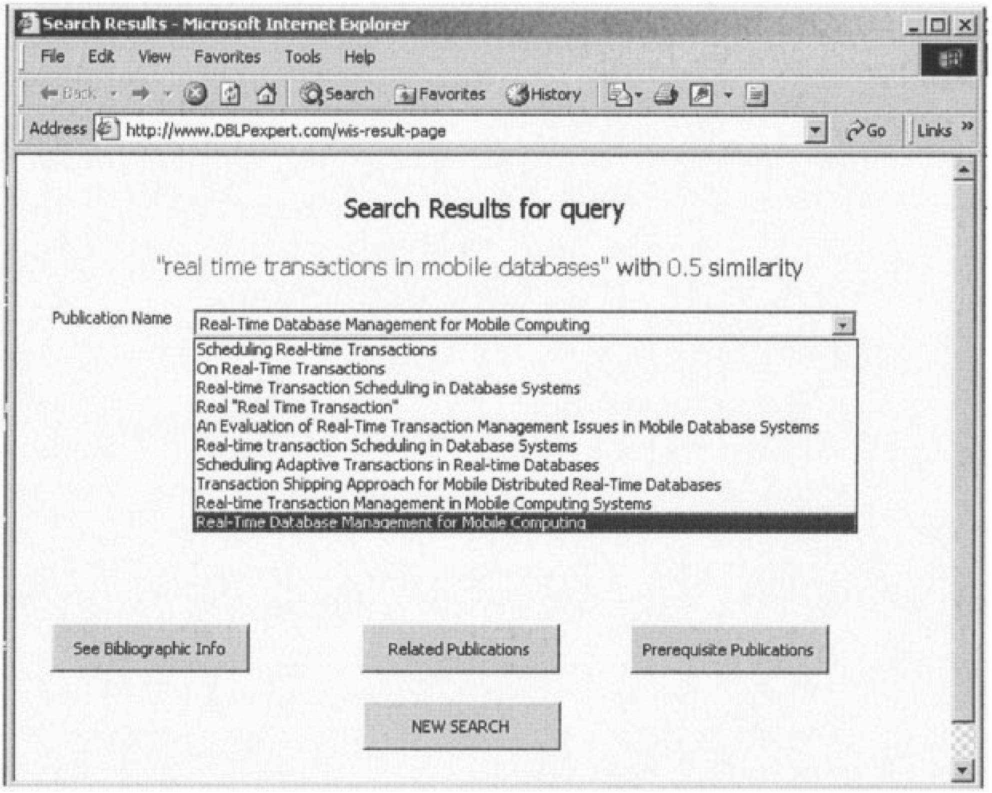
> For the purposes of this paper, we say that such a relationship holds if the topic names of two topics (papers) are similar to each other (computed by vector space model) by more than, say, 50%. Prerequisite metalink holds between two topics T1 and T2 (i.e., T1 → *Prerequisite* T2 if (i) they have at least one common author, they have the *RelatedTo* relationship, and the publication date of paper T1 is earlier than T2, or (ii) the two papers have no common author but all conditions of (i) are satisfied and topic name similarity of T1 and T2 is greater than 80%.

Note that importance scores for topics and metalinks are usually specified in the closed form as discussed in the "Attaching Importance Scores to Metadata Entities" section. For instance, we assign importance score to a topic of type "researcher" according to the number of this researcher's published articles, their citation count, etc. The *RelatedTo* metalink instances are assigned the degree of similarity of the topic names (paper titles) of the participating topics as their importance score. Importance scores of *Prerequisite* instances are computed as the weighted sum of similarity degrees of the participating paper titles, common author percentages of the papers, etc. When there is no intuitive approach for specifying such closed form functions, we use ad-hoc semiclosed form functions (e.g., assign an importance score of 0.9 to all papers published in IEEE TKDE).

The metadata extracted as described in the above is stored in a commercial DBMS (Microsoft SQL Server), with more than 1 million topic and metalink instances. On this expert advice repository, now we can pose sophisticated queries. The prototype application first retrieves topics (papers) that have topic names (titles) similar to a given title above a user-specified threshold, i.e., name-similarity-based matching, as shown in Figure 7. Then, for each of these papers returned in the output, the users can reach the papers that are related to or prerequisite to them. Figure 8 presents papers with names similar to "Real time transaction in mobile databases," where the user can see the prerequisites of the highlighted paper in the list by just clicking on the corresponding button. Note that such queries can be expressed in SQL with some extensions, which we discuss elsewhere along with the query processing algorithms (Özsoyoğlu et al., 2002).

Conclusion

In this study, we develop a Web information space model to allow sophisticated queries/searches over the Web resources. The proposed model has three major components: (i) information resources that are representing the Web-based documents, (ii) expert advice repository model, which constitutes metadata over the resources, and (iii) personalized information model that captures the user preferences and knowledge. Expert advice repositories that are defined by domain experts serve as a semantic index as they identify the topics and their relationships in a resource set, and provide links to the actual occurrences of topics in these resources. The repositories are used for enhanced searching/


Search Results - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://www.DBLPexpert.com/wis-result-page
Go Links
See Bibliographic Info
Related Publications
Prerequisite Publications
NEW SEARCH

Search Results for query

"real time transactions in mobile databases" with 0.5 similarity

Publication Name: Real-Time Database Management for Mobile Computing

- Scheduling Real-time Transactions
- On Real-Time Transactions
- Real-time Transaction Scheduling in Database Systems
- Real "Real Time Transaction"
- An Evaluation of Real-Time Transaction Management Issues in Mobile Database Systems
- Real-time transaction Scheduling in Database Systems
- Scheduling Adaptive Transactions in Real-time Databases
- Transaction Shipping Approach for Mobile Distributed Real-Time Databases
- Real-time Transaction Management in Mobile Computing Systems
- Real-Time Database Management for Mobile Computing

FIG. 8. Result page.

querying of the underlying Web resources, and query outputs are further refined with personalized information.

We make the practical assumption that the Web resource domains associated with expert advice repositories do not span the Web, although they may be arbitrarily large. For the XML Web resources associated with DTDs, we outline an automated approach for building such metadata repositories for practically large subnets and propose methods to match metadata entities defined by different domain experts. Finally, a prototype application that creates an expert advice repository for DBLP Bibliography domain to provide sophisticated querying facilities is presented.

In our recent work (Özsoyoğlu et al., 2002), we present SQL extensions and a query algebra to employ the Web information space model in Web querying context. The extensions are intended to exploit topic/metalink-based metadata to its greatest extent and provide features like text-similarity-based joins and topic closure computations. Future work includes development of a complete rule-based system to create metadata repositories for various actual Web resource sets and a query interface operating on multiple expert advice repositories and user profiles to allow sophisticated querying features provided by our SQL extensions.

Acknowledgment

We wish to thank Mustafa Kutlutürk for implementing the prototype application of our system.

References

Agrawal, R., & Wimmers, E.L. (2000). A framework for expressing and combining preferences. In W. Chen, J.F. Naughton, & P.A. Bernstein (Eds.), Proceedings of the ACM SIGMOD 2000 (pp. 297–306), Dallas, TX: ACM.

Altingövde, I.S., Özel, S.A., Ulusoy, Ö., Özsoyoğlu, G., & Özsoyoğlu, Z.M. (2001a). SQL-TC: A topic-centric query language for web-based information resources (Tech. Rep. No. BU-CE-0108). Ankara, Turkey: Bilkent University, Computer Engineering Department.

Altingövde, I.S., Özel, S.A., Ulusoy, Ö., Özsoyoğlu, G., & Özsoyoğlu, Z.M. (2001b). Topic-centric querying of web information resources. In H.C. Mayr, J. Lazansky, G. Quirchmayr, & P. Vogel (Eds.), Proceedings of database and expert systems applications (DEXA '01) (pp. 699–711), Munich, Germany: Springer Verlag.

Barfourosh, A.A., Nezhad, H.R.M., Anderson, M.L., & Perlis, D. (2002). Information retrieval on the World Wide Web and active logic: A survey and problem definition. Retrieved 2002, from citeseer.nj.nec.com/barfourosh02information.html

Berners-Lee, T. (2000). Semantic Web roadmap (W3C draft). Retrieved 2001, from http://www.w3.org/DesignIssues/Semantic.html

Biezunski, M. (2001). Topic maps at a glance. Retrieved 2001, from http://www.infoloom.com/tmsample/bie0.htm

Biezunski, M., Bryan, M., & Newcomb, S. (Eds). (1999). ISO/IEC 13250, topic maps. Retrieved 2000, from http://www.ornl.gov/sgml/sc34/document/0058.htm

Bray, T., Paoli, J., Sperberg-McQueen, C.M., & Maler, E. (2000). Extensible Markup Language (XML) 1.0 (2nd ed.). Retrieved 2000, from http://www.w3.org/TR/REC-xml

Chritophides, V. (2000). Community Webs (C-Webs): Technological assessment and system architecture. Retrieved 2002, from http://citeseer.nj.nec.com/christophides00community.html

Comer, D. (1979). The ubiquitous B-tree. ACM Computing Surveys, 11(2), 121–137.

ECDL workshop on the semantic web, September 21, 2000, Lisbon, Portugal.

Faloutsos, C. (1985). Access methods for text. ACM Computing Surveys, 17(1), 49–74.

Florescu, D., Levy, A., & Mendelzon, A. (1998). Database techniques for the World-Wide Web: A survey. ACM SIGMOD Record, 27(3), 59–74.

Folch, H., & Habert, B. (2000). Constructing a navigable topic map by inductive semantic acquisition methods. Extreme Markup Languages 2000. Retrieved 2001, from www.limsi.fr/Individu/habert/Publications/Fichiers/folch-et-habert00/folch-et-habert00.html

Garshol, L.M. (2001). Topic maps, RDF, DAML, OIL: A comparison. Retrieved 2002, from http://www.w3.org/TR/1998/NOTE-XML-data-0105/

Giles, C.L., Bollacker, K., & Lawrence, S. (1998). CiteSeer: An automatic citation indexing system. Digital Libraries 98 - The Third ACM Conference on Digital Libraries (pp. 89–98), Pittsburgh: ACM Press.

Horrocks, I. (2002). DAML+OIL: A description logic for the Semantic Web. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 25(1), 4–9.

Kobayashi, M., & Takeda, K. (2000). Information retrieval on the Web. ACM Computing Surveys, 32(2), 144–173.

Kuflik, T., & Shoval, P. (2000). Generation of user profiles for information filtering-research agenda. In Proc SIGIR 2000, 313–315.

Lacher, M.S., & Decker, S. (2001). On the intergration of topic m^1←paps and RDF Data. In Proc Semantic Web Working Symposium 2001.

Layman, A., Jung, E., Maler, E., Thompson, H.S., Paoli, J., Tigue, J., Mikula, N.H., & De Rose, S. (1998). XML Data. Proposal for a Standard. Available at http://www.w3.org/TR/1998/NOTE-XML-data-0105/.

Ley, M. (2001). DBLP bibliography. Retrieved 2001, from http://www.informatik.uni-trier.de/~ley/db/

Magkanaraki, A., Karvounarakis, G., Anh, T.T., Christophides, V., & Plexousakis, D. (2002). Ontology storage and querying (Tech. Rep.). Foundation for Research and Technology Hellas, Institute of Computer Science: Helas.

Mendelzon, A., Mihaila, G., & Milo, T. (1997). Querying the WWW. International Journal on Digital Libraries, 1(1), 54–67.

Microsoft Developers Network Online Support. Retrieved from http://support.microsoft.com/servicedesks/msdn

Mihaila, G.A., Raschid, L., & Tomasic, A. (2002). Locating and accessing data repositories with WebSemantics. VLDB Journal, 11, 47–57.

Online Collections of the Smithsonian Institution. Retrieved from http://www.si.edu

Ontopia-Topic Map Company. tmproc: A topic maps implementation. Retrieved from http://www.ontopia.net/software/tmproc/index.html

Özsoyoğlu, G., Al-Hamdani, A., Altingövde, I.S., Özel, S.A., Ulusoy, Ö., & Özsoyoğlu, Z.M. (2002). Sideway value algebra for object-relational databases. In Proceedings of the VLDB conference 2002 (pp. 59–70), Hong Kong, China: Morgan Kaufmann.

Pepper, S. (1999). Euler, topic maps, and revolution. Retrieved 2001, from http://www.infoloom.com/tmsample/pep4.htm

Rath, H., & Pepper, S. (1999). Topic maps at work. In C.F. Goldfarb & P. Prescond (Eds.), XML handbook (2nd ed.) (Chapter 1). Englewood Cliffs, NJ: Prentice Hall.

Resource Description Framework (RDF) model and syntax specification. (1999). W3C recommendation. Retrieved 2000, from http://www.w3.org/TR/REC-rdf-syntax/

Salton, G. (1989). Automatic text processing. Reading, MA: Addison-Wesley.

Techquila. Topic map samples. Retrieved 2001, from http://www.techquila.com/tm-samples.html

TREC (Text REtrieval Conference) home page. Retrieved from http://trec.nist.gov/

Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., & Nevill-Manning, C.G. (1999). KEA: Practical automatic keyphrase extraction. In N. Rowe & E.A. Fox (Eds.), Proceedings of the fourth ACM conference on digital libraries (pp. 254–255), Berkeley, CA: ACM Press.

WordNet. (1998). Retrieved 2002, from http://www.cogsci.princeton.edu/~wn/

XML Special Issue. (2002). Journal of the American Society for Information Science and Technology, 53(6).

XML Topic Maps (XTM) 1.0. (2001). Retrieved 2001, from http://www.topicmaps.org/xtm/1.0