Topic-Centric Querying of Web Information Resources†

İ.S. Altıngövde[1], S.A. Özel[1], Ö. Ulusoy[1], G. Özsoyoğlu[2], and Z.M. Özsoyoğlu[2]

[1]Department of Computer Engineering, Bilkent University, Ankara, Turkey
(ismaila, selma, oulusoy)@cs.bilkent.edu.tr
[2]Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, Ohio 44106
(tekin, meral)@eecs.cwru.edu

Abstract. This paper deals with the problem of modeling web information resources using expert knowledge and personalized user information, and querying them in terms of topics and topic relationships. We propose a model for web information resources, and a query language SQL-TC (Topic-Centric SQL) to query the model. The model is composed of web-based information resources (XML or HTML 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 as to which expert advice they would like to follow, and which to ignore, etc).

The query language SQL-TC makes use of the metadata information provided in expert advice repositories and embedded in information resources, and employs user preferences to further refine the query output. Query output objects/tuples are ranked with respect to the (expert-judged and user-preference-revised) importance values of requested topics/metalinks, and the query output is limited by either top n-ranked objects/tuples, or objects/tuples with importance values above a given threshold, or both.

1 Introduction

The web today hosts very large information repositories containing huge volumes of data. However, due to the lack of a centralized authority governing the web and a strict schema characterizing the *data* on the web, finding relevant information on the web is a major struggle.

We propose using metadata for an improved searching/querying paradigm. To illustrate our approach, assume that we want to locate movies listed at www.movie-bank.com, related to the novel "Carrie", written by the novelist Stephen King, and are rated at least "very good" (i.e., with an importance value above 0.7 in a scale of 0 to1) by the movie critic (expert) Joe Siegel. Presently, such a task can be performed by browsing the movie pages or by a keyword-based search on a web search engine followed up by a lookup of (some of) the resulting hits, which may be ineffective as well as time-inefficient. Assume that we have an expert that provides a data model for this web site, where "novel", "Carrie", and "Stephen-King" are topics, *RelatedTo* and *WrittenBy* are relationships among topics (called *associations* in the topic map standard, and, in this paper, referred to as *topic metalinks*), and for each topic, there are, perhaps X-Pointer-like pointers pointing to web documents containing "occurrences" of that topic, called *topic sources.* Then, we could formulate and

† This research is supported by a joint grant from TÜBİTAK (grant no. 100U024) of Turkey and the National Science Foundation (grant INT-9912229) of the USA.

H.C. Mayr et al. (Eds.): DEXA 2001, LNCS 2113, pp. 699–711, 2001.

evaluate the query "find movies *RelatedTo* novel Carrie, *WrittenBy* Stephen King, and rated above 0.7 by Joe Siegel" against the data model of the information source, and satisfy the user's request.

In this paper, we describe a "web information space" data model for web information resources, and the query language SQL-TC, where TC stands for topic-centric, to query the data model and web information resources in an integrated manner. The information space is composed of: *i*) Web-based *information resources*, which are XML [6] or HTML documents. *ii*) Independent *expert advice repositories* that contain domain expert-specified model of information resources. We assume that the expert advice, modeled as topic maps, is stored and maintained as XTM [22,23,24] documents. *iii*) *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 maintain user profiles as XML documents.

In this model, topics and topic metalinks are the fundamental concepts through which information resources are modeled and queried. It is important to note that the expert advice repository is a *metadata* model, designed independently from the associated information resources (with the exception of topic source specifications) to model possibly multiple information resources, and capturing the expertise of a domain expert in a lasting manner. Therefore, 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. Finally, SQL-TC query output objects/tuples are ranked with respect to the (domain-expert-judged and user-preference-revised) importance values of requested topics/metalinks. The SQL-TC query output sizes are kept small by returning either (a) top n importance value-ranked objects/tuples, or (b) objects/tuples with importance values above a given threshold, or (c) both.

Thus, the main advantages of our proposal for web search and querying are (a) incorporating expert advice and personalized information, and (b) controlled delivery of query outputs in terms of top-ranking objects/tuples above a given importance value threshold. The disadvantage is the cost of creating and maintaining expert advice and personalized user information. Note that the expert advice, being stable over time, is a one-time effort to create, amortized by its use over time and fast response to user queries.

The query language SQL-TC allows users to query both expert advice repositories, and the associated information resources. Thus, querying resources with respect to multiple expert advices, coupled with the incorporation of personalized information, is expected to produce highly relevant and semantically related responses to users' queries within short time spans.

Next section discusses topic maps and the related standard. Section 3 is devoted to the web information space model with expert advice and user profiles. In Section 4, SQL-TC query language syntax and its features are covered, along with a number of examples.

2 Related Work

We summarize the Topic Map data model, as described in [3, 5, 8, 11, 15]. 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. Topics are *typed,* (e.g., type of the topic Rome is city), and have names. Topic names are also typed; e.g., base name (required), display name (optional), etc. Topic names have *scopes,* e.g., language, style, domain, etc. Topics have *occurrences* 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. Moreover, each occurrence is typed using the notion of *occurrence role.* A *topic association* specifies a relationship between two or more topics. For example, topic Rome *is-in* topic Italy; topic Tom Robbins *was-born-in* topic USA, etc. 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. A number of topic map examples and applications are provided in [4, 9, 13, 16, 17, 18, 19, 20].

XTM (XML Topic Map) is an effort to represent topic maps as XML documents. The proposals and DTDs for XTM are publicly available in [22, 23]. An XTM Processing Model is provided in [24].

3 Web Information Space Model

The three components of the model are information resource model, expert advice model, and user profile model.

3.1 Information Resource Model

Information resources are web-based documents, containing multimedia data of any arbitrary type. For the purposes of this research, we assume that information resources are in the form of XML/HTML documents.

Topic source represents an occurrence of a topic within an information resource. For example, the topic (with name) "Van Gough" occurs multiple times as HTML documents within the documents of the information resource "Online Collections of the Smithsonian Institution" [14], and each such HTML document occurrence constitutes a topic source. 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).

3.2 Expert Advice Model

3.2.1 Topic and Topic Source Reference Entity Types

We assume that the experts in the web information space model are registered and known either through the user profiles or specified in each query explicitly. Each domain expert E_i, $1 \leq i \leq n$, models an information resource in terms of topic and topic source reference entities and, metalink relationships. We start with the topic entity, which constitutes metadata, and 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 [22]) in information resources. Example topic names are "database" (a keyword) and "United Nation's International Policies" (a phrase). Topic names are defined by domain experts, and can be arbitrarily specified phrases or words. Therefore, the issue of similarity between topic names is addressed. To check for the similarity of two topics on the basis of their names, we employ SimTName( ) function, which returns the name similarity of two topics with arbitrarily long topic names as a real value within the range [0, 1].

- *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 "Hamlet" is of type "character" in the domain of "plays". The topic "Paris" may be of type "Greek god" in the domain of "mythology", whereas it is of type "city" in the domain "geography". And, the topic "diabetes" may be of type "chronic disease" in the domain of "medicine". 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 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. As an example, assume that levels 1, 2 and 3 denote levels “beginner”, “intermediate”, and “advanced”. Note that the detail level value of a topic source must be less than or equal to the maximum detail level attribute of the topic.
- *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.
- Topics also have other attributes such as roles, role-playing, etc.

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

The expert E_i, $1 \le i \le n$, states his/her advice on topics as a Topic-Advice function TAdvice() that assigns an *importance value* to topics from one of $[0, 1] \cup \{$No, Don’t-Care$\}$. The importance value is a measure for the importance of the topic, except for the cases below.

a) When the value is “No”, for the expert, the topic is rejected (which is different than the importance value of zero in which case the topic is accepted, and the expert attaches a zero value to it), and
b) When the importance value is “Don’t-Care”, the expert does not care about the use of the topic (but will not object if other experts use it), and chooses not to attach any value to it. Don’t-Care value is used when merging multiple expert advices.

Example 1. Assume that the expert E assigns the following topic advice:
TAdvice(E, TType=”Diabetes”, TName=“*Diabetes Surgeries*”, TDomain=“New Patient Training”) = 0.3
where * denotes a wildcard character that matches any string. This topic advice states that, for training new patients, a topic of type diabetes and with a name containing the phrase “Diabetes Surgeries” is of low importance value.

For the topic advice function TAdvice(), we use the Closed World Assumption with the “No” (or the “Don’t’-Care”) option, denoted as CWA-No (or CWA-Don’t’-

Care) that states that any TAdvice() choice that is not explicitly specified has the value "No" (or "Don't-Care", respectively).

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 document that contains the topic source.
- *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. The detail levels are ordered using the same ordering of the corresponding topics in the attribute Topics.
- Other attributes such as *Mediatype, Role, Last-Modified, Last-Visited* etc.

The expert E_i states his/her advice on topic sources as a Source-Advice function SAdvice() that assigns an importance value to topic sources from one of $[0, 1] \cup \{No, Don't\text{-}Care\}$.

In addition to comparing topic entities by their names (as strings), we compare topics by their topic sources using the function SimTopicSource( ), which returns the similarity of two topics by their topic sources as a real value within the range [0, 1].

3.2.2 Metalink Types and Topic Closures

Topic Metalinks represent relationships among topics. Metalink attributes include types, roles, domains, etc. As an example, consider learning-related metalink type *Prerequisite*, and the metalink instance "Diabetes Complications^2 → *Prerequisite* Diabetes^1" stating that "The prerequisite to understanding/learning the topic Diabetes Complications at level 2 is an understanding of the topic Diabetes at level 1 (or higher). The notation →*Prerequisite* represents an instance of the metalink type *Prerequisite.* Within the context of electronic books, we gave [12] a sound and complete set of axioms for the *Prerequisite* relationship. Any relationship involving topics deemed suitable by an expert in the field can be a topic metalink. For instance, *SubTopicOf* and *SuperTopicOf* metalink types together would represent a topic composition hierarchy.

Metalinks represent relationships among topics, not topic sources. Therefore, they are "meta" relationships, hence our choice of the term "metalink". And, metalink types are usually recursive relationships.

The expert E_i states his/her advice (i) on metalink type signatures as the set *Metalinks,* and (ii) on metalink instances as a Metalink-Advice function MAdvice() that assigns an importance value to a metalink from one of $[0, 1] \cup \{No, Don't\text{-}Care\}$. E_i.Metalinks denote the set of metalink instances (of possibly different types) defined by the expert $E_{i.}$ Similarly, E_i.Topics denote the set of topics defined by the expert E_i.

Example 2. Assume that the expert E states the following metalink signatures: E.Metalinks = {*RelatedTo:* topic → topic, *Prerequisite*: SetOf topic→ SetOf topic } where the first signature states that the *RelatedTo* metalink type takes two topics of any type as arguments, and the second signature states that the *Prerequisite* metalink type takes two sets of topics of any type as arguments. For instance, the expert E states the following metalink instance as an advice: MAdvice (E, Diabetes Care^1 →*RelatedTo* Diabetes Complications^1) = 0.8

This metalink states that the importance value of the metalink "the topic Diabetes Care at the beginner level (1) is related to Diabetes Complications at the beginner level" is reasonably high (0.8) (There may be other causes for diabetes complications).

We assume that different experts specify (a) possibly different topic entities with similar names, (b) overlapping topic sources, and (c) possibly different metalink types and instances. Thus, the system may need to merge the advices from multiple experts and resolve conflicts among advices. An example illustrating this situation along with a user preference-based solution attempt is provided in Example 6 of Section 4.

In this work, we assume that the expert advice described here may either be embedded in information resources or stored independently; in which case, we assume that the expert advice is in the form of an XTM document. A prototype system is developed (but not reported here, due to space considerations) using XTM documents as expert advice repositories.

As stated before, metalink types are usually recursive. For example, *RelatedTo* is both transitive and reflexive. *IsIn* is transitive, but not reflexive; *SubTopicOf* is transitive. Therefore, when a user lists a set X of topics, and asks for topic sources of topics in X as well as others that are *RelatedTo* topics in X, we need to take the "topic closure" of the topic set X with respect to the recursive metalink type *RelatedTo.* We emphasize the notion of *Topic Closures* with respect to recursive metalink types, in order for queries to return results that satisfy all the axioms of the associated metalink types. Given a set X of topics, the query response will include the topic closure X^+, which is formed of all topics that are logically implied by the initial set X.

Clearly, computing topic closures requires a sound and complete set of axioms for the metalink types deployed by the expert E, and a polynomial-time algorithm that computes the topic closure using the axioms. As an example, in our earlier work [12], we gave a sound and complete axiomatization for the *Prerequisite* metalink type. For each new metalink type added into the expert advice model, sound and complete axioms for all metalink types, including those that apply to multiple metalink types are found. To illustrate this, consider the *RelatedTo* metalink type and the cyclic and nondecomposable *Prerequisite* metalink type. Note that, from its signature, all *RelatedTo* metalink instances have a single topic in the LHS and the RHS. Then we have the following axioms:

RelatedTo Axioms:

- Reflexivity. If A →*RelatedTo* B then B →*RelatedTo* A
- Transitivity. If A →*RelatedTo* B and B →*RelatedTo* C then A→*RelatedTo* C

Prerequisite Axioms: Armstrong's axioms.

RelatedTo and *Prerequisite* mixed axioms:

- If X →*Prerequisite* A and A →*RelatedTo* B then C →*RelatedTo* B, $\forall$ C where C $\in$ X.
- If X → *RelatedTo* A and A →*Prerequisite* B then C →*RelatedTo* B, $\forall$ C where C $\in$ X.

With these axioms, one can find the topic closure X^+ of a set X of topics by using the O (n.t) closure algorithm, where n is the number of *Prerequisite* and *RelatedTo* metalinks, and t is the max length of the encoding for a *Prerequisite* or a *RelatedTo* metalink.

3.3 Personalized Information Model: User Profiles

3.3.1 User Preferences

Along the lines of [1], the user U specifies his/her preferences as an ordered set of Accept-Expert, Accept-Expert-Metalink-Importance-Threshold, etc. statements. For the sake of saving space, we illustrate preference functions with an example.

Example 3. Assume that we have three experts W-Clinton, A-Gore, and GW-Bush. The user John-Doe specifies the following preferences:

Expert (John-Doe) = <GW-Bush, W-Clinton>
(Accept the advices of GW-Bush and W-Clinton; reject any advice from A-Gore)

TImportance (John-Doe) = {(GW-Bush, 0.5), (W-Clinton, 0.9)}
(Accept the topics from GW-Bush if GW-Bush-assigned importance is above 0.5; accept the topics from W-Clinton if W-Clinton-assigned importance is above 0.9)

MImportance (John-Doe) = {(W-Clinton, 0.9)} (Always accept the metalinks from GW-Bush; accept the metalinks from W-Clinton if W-Clinton-assigned importance is above 0.9)

SImportance (John-Doe) = {(GW-Bush, 0.5)} (Always accept the sources from W-Clinton; accept the sources from GW-Bush if GW-Bush-assigned importance is above 0.5)

Reject-Topic(John-Doe)={name=”*Lewinski*”,<W-Clinton, Name=”Gift-Taking”>}
(Always reject topics with names containing the word “Lewinski” (regardless of the expert); reject advice from W-Clinton on a topic with name “Gift-Taking”)

Reject-Source (John-Doe) = {Web-Address=www.dirtypolitics.com}

Conflict-Resolution = Ordered-Accept (Follow the order as specified by “expert”: always accept the advice of GW-Bush; accept the advice of W-Clinton only when it does not conflict with the advice of GW-Bush. The other alternative choices include “Accept-and-Merge-All-Advice”)

3.3.2 User Knowledge

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. 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.

Besides detail levels, we also keep historical information for each topic source that the user has visited, which include web addresses (URLs) of topic sources, first/last visit dates and the number of times the source is visited. We use the information on user’s knowledge while evaluating query conditions and computing topic closures, in order to reduce the size of the information returned to the user. In the absence of a user profile, the user is assumed to know nothing about any topic. See Appendix B for an example.

4 Topic-Centric Query Language: SQL-TC

We specify the syntactic constructs of SQL-TC. The formal syntax in an extended Backus-Naur format is given in [2].

select [*topic {.attribute}* | *metalink {.attribute}*] **as** *T* **from resources** *XML: url1,*

using experts *Topic Map1: url1, ...* **as** E1, ... **with user profile** *XML: URL* **as** *U*
where (i) conditions on topics and metalinks of experts; (ii) content-based conditions on sources, (iii) conditions on user profile information
order by [topic] **importance**
stop after n **most important| when importance below** m
| **after** n **most important and when importance below** m

Variables are prefixed by the $ symbol, constants are in quotes, and metalinks are in italics. **Stop after** clause is adapted from [7].

4.1 Querying Web-Based Information Resources

We assume that we have two experts whose advices are at www.sql-tc.com/king.xtm (expert E1) and horror-books.com/books.xtm (expert E2), respectively. The information resources are at www.stephenkinglibrary.com and www.stephen-king.net. As the expert advice and the user profile information, we use the instances provided in Appendices A and B, respectively.

Example 4. (*Topic and source variables, and detail levels*) Using only the advice at www.sql-tc.com/king.xtm, find two highest-ranked novels that are written by the novelist Stephen King, and the novels' detail level 4 reviews from the two information resources.

select [$topic.name, $sourceRef.web-address] **as** T
from resources www.stephenkinglibrary.com, www.stephen-king.net
using experts www.sql-tc.com/king.xtm **as** E
where *WrittenBy* **in** E.Metalinks **and**
$topic = **any** (*WrittenBy* ("Stephen King", “novelist”, “literature”, E)) **and**
$sourceRef = **any** SourceOf($topic, 4, E) **and** “review” **in** $sourceRef.roles
order by $topic **importance**
stop after 2 **most important**

Novel names are topic names, and the novel reviews constitute topic sources. The result of the query is a 2-column table with 2 tuples. The first atomic formula in the where clause states that *WrittenBy* is a metalink type declared by expert E. Assume that the metalink type *WrittenBy* has the signature: *WrittenBy*(E): SetOf author → novel

In the second where clause statement, the variable $topic is instantiated by one of the novel entities returned by the *WrittenBy()* metalink where each selected novel is authored by the topic that has TName of “Stephen King”, TType of “novelist” and TDomain of “literature”, and specified by the expert E. This query illustrates two types of variables, namely, $topic which is a topic variable, and $sourceRef which is a topic source reference variable. SourceOf() is a function that takes in the triple <topic entity, detail level, expert>, and returns a set of topic source reference (TSRef) entities at the given detail level as specified by the given expert. Thus, in the above query, the value of $sourceRef.web-address expression is, according to expert E, the web addresses of topic sources at detail level 4 obtained from the topic reference entities for the topic $topic.

Using the expert advice in Appendix A, this query produces 4 tuples; however, only the two highest ranked tuples (one for Carrie with importance value of 1, and another for the Stand with the importance value of 0.8) are returned as shown in Table1.

Table 1. Output of the SQL-TC query in Example 4

Tname	SourceRef.Web-address
“Carrie”	www.critics.com/carrie.html
“The Stand”	www.critics.com/stand.html

Example 5. (*Topic closure computation and user profiles*) Using only the advice of expert E and excluding the novels read by the user, find the highest ranked novel and its detail level 4 reviews where the novel is written by Stephen King and related to the novel “Wizard and The Glass”.

```
select [$topic.name, $sourceRef.web-address] as T
from resources www.stephenkinglibrary.com, www.stephen-king.net
using experts www.sql-tc.com/king.xtm as E
with user profile www.myprofile.com
where  WrittenBy, RelatedTo in E.Metalinks and
          $topic = any (WrittenBy ("Stephen King", ”novelist”, ”literature”, E)
                         and RelatedTo* (“Wizard and The Glass”, ,”literature”, E)) and
          $sourceRef = any SourceOf($topic, 4, E)  and  “review” in $sourceRef.roles
                                                                                  and
          $topic not in GetTopics(U.UserKnowledge)
order by importance
stop after 1 most important
```

We assume for this query that the metalink type *RelatedTo* of expert E has the signature *RelatedTo*(E): novel → novel. Note that in this query the user asks for the highest-valued tuple, not the highest-valued novel. Derived importance value computation of output tuples [2] takes place, and the tuple in Table 2 is chosen. Let us discuss the interpretation of this query using the expert advice repository and user profile instances in the Appendices. The novels that are related to the novel “Wizard and The Glass” are recursively located. From Appendix B, the output returns only those novels that are not known by the user. For instance, according to the expert advice in Appendix A, the topics that are related to “Wizard and The Glass“ are “The Wasteful Lands”, “Drawings of Three” and “Dark Tower”. However, since the novel “Dark Tower” is already known according to the user profile (given in Appendix B), it is not included in the final result, and the tuple (NOT the novel) with the highest importance value is selected.

Table 2. Output of the SQL-TC query in Example 5

Tname	SourceRef.Web-address
"The Wasteful Lands"	www.critics.com/dark3.html

Example 6. (*User preferences, user knowledge and multiple experts*) Using first the expert www.sql-tc.com/king.xtm, and then, if there are no conflicts, the expert www.horror-books.com/books.xtm, find all novels and their summaries such that the main characters of the selected novels are influenced from "Jack Park", the main

character of the novel “The Stand”, and retrieve only those sources that have not been visited by the user in the last 30 days.

select [$topic.name, $sourceRef.web-address] **as** T
from resources www.stephenkinglibrary.com, www.stephen-king.net
using experts www.sql-tc.com/king.xtm **as** E1, www.horror-books.com/books.xtm **as** E2
with user profile www.myprofile.com as U
where *NovelsOfNovelCharacters*, *InfluencedBy* **in** (E1, E2). Metalinks **and**
$topic=**any** *NovelsOfNovelCharacters* (
*InfluencedBy** ("Jack Park", hero, novel characters,),) **and**
$sourceRef = **any** SourceOf($topic, ,) **and** “summary” **in** $sourceRef.roles **and**
$sourceRef.web-address **in** GetSourceAddresses (U.UserKnowledge) **and**
GetLastVisitedDays (U.UserKnowledge, $sourceRef.web-address) > "30"

The second where clause assigns a novel to the topic variable $topic where the novel has a main character influenced by “Jack Park”, in the domain of “literature”. For both experts, we assume that the signatures of the metalink types *InfluencedBy* and *NovelsOfNovelCharacters* are the same, and each is defined as

InfluencedBy (E): novel-character → novel-character and
NovelsOfNovelCharacters (E): novel-character → SetOf novel

where E denotes either of the two experts. Note that, in the query, the selection of the expert for the above metalinks (and the expert of the function SourceOf()) is not specified in the query, and deferred to the user’s preferences. Also, in the SourceOf() function, a topic source at any detail level is accepted.

For this example, we assume that the *InfluencedBy* metalink is binary, transitive, and cyclic, and we apply the corresponding topic closure computation algorithm for this case. According to the advice of expert www.sql-tc.com/king.xtm (E1 in Appendix A), “Jack Park”, influences the character “John Smith”. As “John Smith” is claimed to be the main character of the novels “Scream” and “Maniac” by expert www.horror-books.com/books.xtm (E2 in Appendix A), the topic closure computation will bind each of “Scream” and “Maniac” to the $topic variable. Thus, $sourceRef.web-addresses will be assigned to the corresponding sources www.books.com/scream.html and www.books.com/maniac.html. The function GetSourceAddresses() returns addresses of visited sources and the function GetLastVisitedDays() retrieves the days since the last-visit of a given source from the user profile database U (in Appendix B). Subsequently, the entire query will return www.books.com/maniac.html as it is the only source that is visited by the user and not in the last thirty days.

Note that this query employs more than one expert advice, and the issue of conflicts among different expert advice comes up. In the user preferences (given in Appendix B), first the advice of E1 and then, if there are no conflicts, the advice of E2 are to be accepted. Assume that the following metalink advice instances are encountered during the topic closure computation with respect to the *InfluencedBy* metalink type:

MAdvice(E1, “John Smith” →*InfluencedBy* “Jack Park”) = 0.8
MAdvice (E2, “John Smith” →*InfluencedBy* “Jack Park”) = “No”

The query evaluation relies first on E1 and includes the character “John Smith” in the closure set, or relies on E2 and discards the character “John Smith” (and thus all other topics that may possibly be added to the closure because of the inclusion of “John Smith”) from the closure. To resolve the conflict, the query engine consults the metalink-importance-threshold statements declared in the user preferences, and discards the advice with a lower importance value than the given threshold. The user preferences (of Appendix B) declare threshold values 0.5 and “Don’t-Care” for experts E1 and E2 respectively. And, the conflict-resolution statement of the user’s preferences declares an ordered acceptance of advices. Thus, we add “John Smith” into the topic closure set.

4.2 Querying Expert Advice Repositories

Example 7. (*Metalink attributes*) Find top 30-ranked metalinks in the domain of literature and having an importance value of at least 0.7 for the expert www.sql-tc.com/king.xtm such that, in each such metalink, Stephen King is a participator.

select [$metalink.type] **as** T
using experts www.sql-tc.com/king.xtm **as** E
where $metalink **in** E.Metalinks **and**
    $metalink**= any** (MetalinksWithTopic ("Stephen King", , , E)) **and**
    $metalink.domain = "literature"
order by importance
stop after 30 **most important and when importance below** 0.7

The function MetalinksWithTopic() takes a topic (either fully identified by TName, Ttype, TDomain, and TAuthor in the given order, or partially identified), and returns metalink instances. According to Appendix A, the query output includes the “*WrittenBy*” metalink type.

References

[1] Agrawal, R., Wimmers, E.L., “A Framework for Expressing and Combining Preferences”, ACM SIGMOD Conf., pp. 297-306, 2000.

[2] Altıngövde, I.S., Ozel, S.A., Ulusoy, O., Ozsoyoglu, G., Ozsoyoglu, Z.M., SQL-TC: A Topic-Centric Query Language for Web-Based Information Resources, manuscript in preparation, 2001.

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

[4] Biezunski, M, "A Topic Map of This Conference's Proceedings", Proc. of GCA, 1996, http://www.infoloom.com/IHC96/mb214.htm

[5] Biezunski, M., Bryan, M., Newcomb, S., editors, ISO/IEC 13250, Topic Maps, available at http://www.ornl.gov/sgml/sc34/document/0058.htm.

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

[7] Carey, M.J., Kossmann, “On Saying “Enough Already” in SQL”, ACM SIGMOD 1997.

[8] ISO 13250 Topic Map Standard available at http://www.w3c.com

[9] Ksiezyk, R., "Answer is Just a Question [of matching Topic Maps], Proc. of XML Europe 2000, GCA, Alexandria, VA, 2000.

[10] Microsoft MSDN Online Support, available at http://support.microsoft.com/servicedesks/msdn.

[11] Newcomb, S., Biezunski, M., "Topic Maps go XML", XML Europe 2000, June 2000.

[12] Ozsoyoglu, G., Balkir, N.H., Cormode, G., Ozsoyoglu, Z.M., "Electronic Books in Digital Libraries", IEEE Advances in Digital Libraries Conf., Washington, D.C., May 2000.

[13] Ontopia Topic Map Technology, available at www.ontopia.net

[14] Online Collections of the Smithsonian Institution, available at http://www.si.edu.

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

[16] Rath, H.H., "Topic Maps Self Control", Extreme Markup Lang. 2000, Montreal, PQ, Canada, 2000.

[17] The K42 Topic Map Engine, available at http://k42.empolis.co.uk

[18] Topic Maps available at www.infoloom.com

[19] Topic Map Samples, available at http://www.techquila.com/tmsamples/

[20] tmproc: A Topic Maps implementation, available at http://www.ontopia.net/software/tmproc/index.html

[21] XML Pointer Language (XPointer) version 1.0, available at http://www.w3.org/TR/xptr/

[22] XML Topic Maps (XTM) 1.0 available at http://www.topicmaps.org/xtm/1.0

[23] XTM: XML Topic Maps- working documents, available at http://www.doctypes.org/xtm/home.html

[24] XTM Processing Model 1.0, TopicMaps.Org AG Review Specification, 4 Dec 2000, available at http://topicmaps.org/xtm/1.0/xtmp1.html

Appendix A: Expert Advice Repositories

In the following, we provide expert advices as list for the ease of illustration. Clearly, the expert advice repositories may be in the form of text files, XML files and/or tables/objects of any conventional databases.

To save space, we only provide topics, sources and metalinks that are illustrated in the examples throughout the paper.

A.1 Expert Advice Provided in www.sql-tc.com/king.xtm (Expert E1)

Each topic of the expert advice is specified in the form of tuple: <Tid, TDetail level, Ttype, Tname, Tdomain, T-Advice, Source>.

Topics (E1) = {<T1, -, novelist, "Stephen King", literature, 1, S1>, <T2, -, novel, "Carrie", literature, 1, {S2, S3}>, <T3, -, novel, "The Stand", literature, 0.8, S4>, <T4, 4, novel, "Wizard and The Glass", literature, 0.3, ->, <T5, 3, novel, "The Wasteful Lands", literature, 0.4, S5>, <T6, 2, novel, "Drawings of Three", literature, 0.6, S6>, <T7, 1, novel, "Dark Tower", literature, 0.8, ->, <T8, -, hero, "Jack Park", novel characters, -, ->, <T9, -, character, "John Smith", novel characters, -, ->}

Similarly, each metalink of the expert advice is specified in the form of tuple: <Mid, Mtype, Mdomain, Antecedent players, Consequent players, M-advice>.

Metalinks (E1) = { <M1, WrittenBy, {literature, horror}, T2, T1, 1>, <M2, WrittenBy, {literature, horror}, T3, T1, 0.6>, <M3, WrittenBy, {literature, horror}, T4, T1, 0.6>, <M4, WrittenBy, {literature, horror}, T5, T1, 0.6>, <M5, WrittenBy, {literature, horror}, T6, T1, 0.6>, <M6, WrittenBy, {literature, horror}, T7, T1, 0.6>, <M7, RelatedTo, -, T7, T6, 0.6>, <M8, RelatedTo, -, T6, T5, 0.5>, <M9, RelatedTo, -, T5, T4, 0.3>, <M10, InfluencedBy, -, T9, T8, 0.8>}

Note that, in the above list, the attributes of tuples may be set-valued. Although we refer the player topics by their internal ids (as in the prototype implementation) for the sake of saving space, the player topics could also be specified by the quadruples of the form TName, TType, TDomain, TDetail-level.

A source element of the expert advice is specified in the form of tuple: <Sid, Web-address, Role, MediaType, LastUpdated, Detail level, S-Advice>.

Sources(E1) = { <S1, www.king.com/, Website, multimedia, 16.01.2001, -, 1>, <S2, www.books.com/carrie.html, Summary, Text, -, -, 0.5>, <S3, www.critics.com/carrie.html, Review, Text, -, 4, 0.8>, <S4, www.critics.com/stand.html, Review, Text, -, 4, 0.7>, <S5, www.critics.com/dark3.html, Review, Text, -, 4, 0.8>, <S6, www.critics.com/dark2.html, Review, Text, -, 4, 0.3>}

A.2 Expert Advice Provided in www.horror-books.com/books.xtm (Expert E2)

Similarly, the metadata that is specified by expert E2 is given in the below.

Topics(E2) ={<T10, -,novel, “Scream”, literature, 0.3, S7>, <T11, -, novel, “Maniac”, literature, 0.4, S8>, <T12, -, hero, "Jack Park", novel characters, -, ->, <T13, -, character, "John Smith", novel characters, -, ->}

Metalinks(E2) = { <M11, InfluencedBy, -, T13, T12, No>, <M12, NovelsOfNovelCharacters, -, T13, T10, 0.6>, <M13, NovelsOfNovelCharacters, -, T13 , T11, 0.2>}

Sources(E2) = {<S7, www.books.com/scream.html, Summary, text, 12.02.2001, -, 0.6>, <S8, www.books.com/maniac.html, Summary, text, 13.02.2001, -, 0.7>}

Appendix B: Personalized Information for User U

In the following, we provide personalized information in terms of user preferences and user knowledge for a typical user U. Assume that user profile is kept in the virtual web location **www.myprofile.com**.

User-Preferences (U) contains a set of statements as follows:
Expert (U) = <www.sql-tc.com/king.xtm, www.horror-books.com/books.xtm>
TImportance(U) = { (www.sql-tc.com/king.xtm, 0.5), (www.horror-books.com/books.xtm, 0.3)}
Mimportance(U) = {(www.sql-tc.com/king.xtm, 0.5), (www.horror-books.com/books.xtm, “Don’t-Care”)}
Simportance(U) = {(www.sql-tc.com/king.xtm, 0.5), (www.horror-books.com/books.xtm, 0.3)}
Reject-S (U) = {www.sking-fanatics.com}
Conflict-R (U) = Ordered-Accept

User-Knowledge (U)

User knowledge is specified in the form of tuple: <TName, DetailLevel, Source-address, Sourcerole, Sourcemediatype, FirstVisit, Last visit,Visit No>
User-Knowledge (U) = {<"Scream", -, www.books.com/scream.html, summary, text, -, 12.02.2001, 2>,
<"Maniac", -, www.books.com/maniac.html, summary, text, -, 13.02.1999, 3>,
<"Dark Tower", 1, www.books.com/dark1.html, review, text1, -, -, 3>}