TOPIC-CENTRIC QUERYING OF WEB RESOURCES

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BİLKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

İsmail Sengör Altıngövde

September, 2001

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Özgür Ulusoy(Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Tuğrul Dayar

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Fazlı Can

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet Baray
Director of the Institute

ABSTRACT

TOPIC-CENTRIC QUERYING OF WEB RESOURCES

İsmail Sengör Altıngövde
M.S. in Computer Engineering
Supervisor: Assoc. Prof. Dr. Özgür Ulusoy
September, 2001

As the world wide web (WWW) has evolved to be almost the largest source of information that is known by human beings, locating relevant information on the web in a reasonably short time has become a major struggle. High quality indices and (sometimes specialized) search engines that employ information retrieval techniques are widely used for keyword based searches, and a number of web query languages have also been developed, mostly for research purposes. However, most of the keyword-based approaches are vulnerable to the noise on the web, leading to unqualified results with lots of irrelevant documents; whereas the web-query languages lack the speed or generality to be used in practical cases.

In this thesis, we make use of *metadata* (along with some XML-based standards) to characterize the web resource domains, and to provide sophisticated querying features with high-quality results and a reasonably fast response time. We propose a "web information space" *metadata* model for web information resources, and a query language SQL-TC (Topic-Centric SQL) to query the model. The web information space 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 (user profiles and preferences, as XML documents). Expert advice is specified using *topics* and *relationships* among topics (called *metalinks*) in a particular domain of interest, along the lines of the recently proposed *topic maps*. Experts also attach importance values to topics and metalinks that they specify, and link them to actual information resources on the web whenever possible, creating a *semantic index* over the resources. User profiles keep track of user knowledge and navigation history in terms of these topics and their (visited) sources, whereas user preferences declare users' attitudes and confidence for the

choices of particular experts.

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. Therefore, the query output of SQL-TC is expected to produce highly relevant and semantically related responses to user queries within short amounts of time.

Keywords: metadata, XML, Topic Maps, web data modeling, web querying, semantic indexing, user profile.

ÖZET

WEB BİLGİ KAYNAKLARININ KONU-MERKEZLİ SORGULANMASI

İsmail Sengör Altıngövde
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Doç. Dr. Özgür Ulusoy
Eylül, 2001

İnternetin insanlık tarafından bilinen hemen hemen en büyük bilgi kaynağı olmasıyla, aranılan bilgiye web üzerinde yeterince kısa bir zamanda ulaşabilmek önemli bir sorun ve araştırma konusu haline gelmiştir. Uzmanlarca hazırlanmış yüksek kaliteli indekslerden ve bilgi erişim tekniklerini kullanan (kimi zaman özelleştirilmiş) arama motorlarından anahtar-kelime tabanlı aramalar için faydalanılırken, çoğu akademik araştırma amaçlı bir çok web sorgu dili de literatürde önerilmiştir. Ancak, anahtar-kelime aramaya dayalı yaklaşımlar web üzerindeki "bilgi kirliliğine" karşı savunmasız olup aranılan konuyla ilgisi olmayan pek çok sonuç geri döndürebilmektedir. Diğer taraftan web sorgu dilleri ise pratikte kullanımlarını sağlayabilecek hız ve genel geçerlikten uzak durumdadır.

Bu tezde, web kaynak alanlarını temsil etmek ve yüksek kaliteli arama sonuçlarını kabul edilebilir sürede geri döndürebilecek sorgu kapasitesi sağlamak için *"metadata"*dan ve beraberinde XML temelli bir takım standartlardan faydalanılmaktadır. Bu tezde web bilgi kaynakları için "web bilgi uzayı" *metadata* modeli ve bu model üzerinde işlem yapan SQL-TC (Konu-merkezli SQL) sorgu dili sunulmaktadır. Bahsedilen web bilgi uzayı modelinin bileşenleri ise web-tabanlı bilgi kaynakları (internetteki XML veya HTML belgeleri), uzman öneri veri-tabanları (bilgi kaynakları için alan/konu uzmanlarınca belirlenmiş metadata) ve kullanıcılar için özelleştirilmiş bilgilerdir (XML belgeleri olarak saklanan kullanıcı profilleri ve tercihleri). Uzman önerileri, literatürde yakın zamanda sunulmuş olan *konu haritaları* çalışması göz önüne alınarak belli bir ilgi alanındaki *"konu"*lar ve konular-arası *"ilişkiler"* (*metalink*'ler) kullanılarak verilmektedir. Alan uzmanları, tanımladıkları konulara ve metalinklere bunların o alandaki önemlerini yansıtan sayısal önem değerleri de iliştirmektedirler. Uygun koşullar

altında, bu konu ve metalinklerin web ortamındaki gerçek bilgi kaynaklarıyla ilişkilendirilmesiyle uzman öneri veritabanı bu bilgi kaynakları için bir *kavramsal indeks* işlevi kazanmaktadır. Modelin bir diğer bileşeni olan kullanıcı profilleri ise kullanıcının bilgi seviyesini ve web dolaşım tarihçesini yine konular ve ziyaret edilen web kaynakları cinsinden saklamaktadır. Kullanıcı tercihleri ise belli bir alan uzmanının önerilerine karşı kullanıcının yaklaşımını ve/veya güvenini yansıtmaktadır.

Bu tezde sunulan SQL-TC sorgu dili uzman öneri veri-tabanlarında sağlanan metadata'yı kullanmakta ve sorgu sonuçlarını daraltmak ve özelleştirmek için (yukarıda bahsedilen) kullanıcı bilgilerinden faydalanmaktadır. Sorgu sonucunda bulunan nesneler veya veri tablosu satırları ise uzmanlarca atanan ve kullanıcı tercihlerine göre değerlendirilen önem değerleri göz önüne alınarak sıralanmaktadır. Bu sıralama sonrasında kullanıcıya (isteğine göre) sadece en önemli ilk n nesne ve/veya belli bir önem değerinin üstündeki nesneler ya da her ikisi birden sunulabilmektedir. Bunlardan dolayı, SQL-TC sorgu dilinin aranılan konuyla "anlamsal" olarak ilişkili sonuçları kısa zamanda bulabilmesi beklenmektedir.

Anahtar sözcükler: metadata, XML, Konu Haritaları, web veri modellemesi, web sorgulaması, kavramsal indeksleme, kullanıcı profili.

Acknowledgement

First of all, I would like to express my deepest thanks and gratitude to my supervisor Assoc. Prof. Dr. Özgür Ulusoy, for encouraging me to step into the academic life and for his invaluable suggestions, support and guidance during the last two years. Without his motivation and patience, this research would not be that much enjoyable for me.

I owe special thanks to Prof. Dr. Gültekin Özsoyoğlu and Prof. Dr. Z. Meral Özsoyoğlu, for their excellent ideas, patient revisions and support, all of which invaluably contributed to this thesis.

I would like to thank Prof. Dr. Fazlı Can and Assoc. Prof. Dr. Tuğrul Dayar, for spending their time and effort to read and comment on this thesis. I would also like to acknowledge the financial support of TÜBİTAK (of Turkey) under the grant 100U024 and NSF (of the USA) under the grant INT-9912229.

I am grateful to my colleague S. Ayşe Özel, for her cooperation during this study and invaluable contributions. I would like to thank my officemate Dr. Yücel Saygın for his moral support and insightful views, and all my friends who were always with me with their technical and moral support.

Finally, I would like to thank my family, for being with me whenever I need, believing in me whatever I do and walking with me wherever I go. Without their incredible love and patience, this thesis would never be completed.

To my family

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Due to the enormous growth of the world wide web (WWW) 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. Researchers in many fields of computer science including databases, information retrieval, data mining and AI investigate the problem of locating only the relevant information and locating it fast.

Extensible Markup Language (XML) [48], one of the recent developments for web and adopted as a standard by the World Wide Web Consortium (W3C), provides a simple syntax for self-describing web data, and will likely become a data exchange model on the web [34]. Meanwhile, the *topic maps* model [44], another recently adopted standard, allows web administrators to define additional semantics for web documents in terms of topics, topic associations, etc., and, thus, provides a new way of navigating information resources. Basically, a topic map is a semantic data model describing the contents of web documents in terms of topics and topic associations, and therefore constitutes a "metadata" model. Thus, topic maps allow web users to benefit from semantic data modeling that may be employed in a variety of ways, one of which is to improve the performance of search engines [75]. As one would anticipate, efforts are underway to combine

XML and topic maps, and the result is the XML topic maps (XTM) model that intends to express a syntactic and functional subset of the topic map standard in XML [75].

To illustrate the advantages of using metadata for an improved searching/querying paradigm, consider the movie database at www.movie-bank.com. Assume that we would like to locate movies listed at this site and are related to the novel "Carrie", written by the novelist Stephen King, and rated at least "very good" (i.e., with an importance value above 0.7 in a scale of 0 to 1) 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 thesis, referred to as *topic metalinks*), and for each topic there are perhaps X-Pointer-like [22] pointers pointing to web documents containing "occurrences" of that topic, called *topic sources*. Then, we could formulate and 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 an efficient manner (assuming that the query optimization takes place). One can thus view the data model as a "knowledge (-based)-index" to the web information resource. Then, such a knowledge-index can be the starting point for a semantic-based web search and controlled delivery of the response to users.

In this thesis, 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 [39]. The information space is composed of:

- Web-based information resources which are XML or HTML documents.
- 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 documents.

- *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 we model and query the contents of information resources. 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. Another implication is that the topic metalinks are more likely to specify recursive relationships between topics, such as the relationships *RelatedTo, LeadsTo, IsIn, Prerequisite, ComposedOf,* etc. 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 advantage 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. We also make the practical assumption that the modeled information resources, however large they may be, do not span the web; they are defined within *subnets* such as the ACM SIGMOD

Anthology sites, or the larger domain of Microsoft Developers Network sites [7], or the very large domain of Online Collections of the Smithsonian Institution [9]. And, different expert advice may be provided for the same web information resource to express varying viewpoints of different domain experts.

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

In this thesis, we make use of *metadata* (along with some XML-based standards) to characterize the web resource domains, and to provide sophisticated querying features with high-quality results and a reasonably fast response time. We propose a "web information space" *metadata* model for web information resources, and a query language SQL-TC (Topic-Centric SQL) to query the model.

In the next chapter, we first briefly summarize XML, topic maps and the related standard, and XTM. Then, we discuss the earlier works in the literature that also exploit metadata for the purposes of web searching/querying. Chapter 3 is devoted to the web information space model with expert advice and user profiles. In Chapter 4, SQL-TC query language syntax and its features are covered, along with a number of examples. Chapter 5 describes the prototype implementation employing XTMs as expert advice repositories. Finally, we conclude and point out future research directions in Chapter 6.

Chapter 2

Background and Related Work

2.1 Tagging the Content on the Web: XML

Extensible Markup Language (XML) [48] is becoming a universal standard for data exchange on the web, recommended by the World Wide Web Consortium (W3C). XML-Data [27] represents 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., Document Type Definitions and valid documents). In this sense, XML was designed to describe the content [34], rather than its presentation. SAX [12] and DOM [29] are two APIs for parsing XML data. It is claimed that the use of XML will bring a major change in the structure of web information [47] and XML will be the basis for data interchange on the Internet, i.e., electronic data interchange (EDI) format [58]. Recent research activity on XML and databases include, among others, storing and querying XML documents, XML views, XML architectures and others. See as an example [28].

2.2 Motivation for Metadata on the Web and Frameworks to Express Metadata

One definition of *superimposed* information (or metadata, as we call throughout this thesis) is the data that is placed over the existing information resources to help organizing, accessing and reusing the information elements in these resources [72]. The need for metadata over the web is justified in [72] with three key observations: (i) the increasing amount of digital information on the web, (ii) emerging mechanisms allowing to address the information objects in a finer granularity, and (iii) the increasing amount of inaccurate and/or worthless information over the web, which is directly related to the first observation. In this sense, the authors identify a requirement for annotation and evaluation of web-based resources to facilitate the access to qualified and relevant resources, and thus to make use of the information on the web to its greatest extent.

As metadata leaves the base information pool untouched and provides alternative views of the base data, the question of attaching metadata to the web resources may be related to the well-known issues of data integration and view creation in the database literature. Although all of these problems have points in common, metadata on the web is said to have -at least- two distinguishing aspects with respect to the views over databases: (i) the metadata associated with the resources may not be explicitly present in the original resource set, and (ii) it does not have a fixed schema, but rather expressed in a semi-structured or unstructured format [72]. It is further claimed in [72] that, traditional data integration methods may be prohibitively costly to provide metadata over large sets of information resources on the web.

Thus, recognizing the need for metadata over the web, we will focus on two emerging and competing standards providing a framework to specify metadata on the web: Topic Maps and RDF, both of which are to be discussed in the upcoming sections, as well as the XML representation of Topic Maps.

2.2.1 Topic Maps Standard

As mentioned in [77], the topic maps standard is an effort to provide a metadata model for describing underlying data in terms of topics and associations, and includes links to actual occurrences of these constructs in the information resources. In the following, we summarize the key concepts of Topic Maps as described in [77, 80, 81].

- *Topic*: A *topic* can be anything whatsoever a person, an entity, a concept, anything regardless of whether it exists or has any other specific characteristics about which anything may be asserted. Thus, a *topic* in the topic map data model is considered as an object (or a handle) that corresponds to a *subject* in the real-world (or in the author's mind). A subject of discourse may be electronically *addressable* (i.e., a file at a particular web address) or not (i.e., "John Lennon" is not an addressable subject). In some sense, a topic in the topic map model reifies the subject, makes the subject "real" for the system, whenever it is not addressable. For instance, in the context of an encyclopedia, a topic might represent subjects such as "Turkey", "Italy", "Rome", "John Lennon" or "love". More specifically, anything that might be an entry (either addressable or non-addressable) in the encyclopedia may be a topic. Now, we can declare assertions for these subjects using the topics that correspond to them and which are obviously addressable in a topic map structure.

- *Topic type*: Every topic has one or more *types*, which contribute to a typical *class-instance* (or IS-A) relationship and they are themselves defined as topics. Along with the lines of the above example, "Turkey" and "Italy" are of type "country", "Rome" is of type "city", "John Lennon" is of type "singer" and "love" is of type "emotion".

- *Topic name*: Each topic may have zero or more names. The standard [44] distinguishes three types of names for a topic: *base name, display name* and *sort name*. The base name for topic "John Lennon" would be as-is, whereas the sort name could be "Lennon, J".

- *Topic occurrence*: An *occurrence* is a link to an addressable information resource that is relevant to the topic. Every occurrence plays a role expressed by the *occurrence role type*, which is again a topic. For example, topic "Turkey" is *described* in an *article*, "Rome" is *depicted* in a *picture*, and "John Lennon" is *mentioned* at *Beatles Fans' web site*. The "article", "picture" or "web site" in this example are occurrences, whereas "describe", "depict" and "mention" denote the occurrence type of topic in the corresponding resource. Note that, in general, actual information resources do not exist within the topic map itself, and an addressing/referencing scheme such as HyTime [1], Xpointers [22] or Xlink [21] is used to refer to an information resource.

- *Topic association*: An *association* describes the relationship between two or more topics. Each association is of a specific *association type*. Each associated topic plays a *role* in the association. The association type and association role type are both topics. The assertion "Rome" *is-in* "Italy" is an association, where the association type is *is-in*, and the role types for the player topics "Italy" and "Rome" are "container" and "containee", respectively. Another association may be "John Lennon" *wasBornIn* "Liverpool", with the association role types "person" and "city".

  Note that the constructs we have described up to this point give way to a very powerful structure allowing us to define topics, which may be *anything* we want to talk about, to make assertions about their interrelationships and to link these topics to actual (addressable) web-resources, if they ever exist. Next, we discuss additional constructs to enrich the semantics of the model.

- *Scope and theme*: Any assignment to a topic is considered valid within certain limits, which may or may not be specified explicitly. The validity limit of such an assignment is called its *scope*, which is defined -again- in terms of topics, so-called, *themes*. For instance, topic "Paris" is of type "city" in the scope of (topic) "geography", whereas "Paris" is of type "hero" in the scope of "mythology".

- *Public subject*: This is a mechanism intended to establish the identity of a

topic, say, in case of merging two topic maps. A public subject is an addressable information resource which unambiguously identifies the subject of topic in question. For instance, the public subject of topic "Topic Maps" may be the address of the ISO document [44] defining the standard officially and publicly. Note that, to encourage the use of topic maps, a number of parties are attempting to provide public subject directories.

In an ideal case, a (consistent) topic map may include at most one topic element representing a real world subject, and when two or more topic map documents are merged, the topics representing the same concept may be deduced and merged as well. The public subject is one mechanism to infer the topic equality; other mechanisms, as well as the issue of merging topic maps, are discussed in the more specific case of XTMs in the next section.

The ISO standard [44] officially defines Topic [Navigation] Map syntax for the interchange of information in terms of the constructs mentioned above and uses Standard Generalized Markup Language (SGML) architectural forms [67] based on HyTime hyperlinks [1]. Basically, a topic map may be seen as (but not restricted to) an SGML (or XML, see the next section) document in which different element types are used to represent the constructs that are described above [80].

The basic motivation behind the emergence of topic maps was the need to be able to merge indexes for different document collections [80]. However, it soon turned out that topic maps would be capable of handling tables of contents, glossaries, thesauri, cross references, etc., and beyond.

The power of topic maps as a navigational-aid comes from the fact that it provides a knowledge index over a set of information resources [83]. Today's search engines are employing full-text indices, whereas topic maps propose a kind of *semantic index*, over the topics and their relationships that exist in the underlying information resources. As it is mentioned in [81], the full-text indexing approaches suffer from the lack of discrimination: they index literally everything and end up in highly noisy query responses. On the other hand, topic maps

capture the structure of the knowledge, promising the opportunity of much sophisticated navigation and querying of underlying resources.

Topic maps may also prove to be useful in the creation and maintenance of information as an organising principle [80]. As it is mentioned above, the occurrences of topics are links to actual information resources that are outside the topic map itself, allowing a clean separation of data into two domains: the domain of topics and their associations (*core metadata*) and the domain of occurrences (*the actual data*, or *information resources*) [77, 80]. The independence of these two domains means that a topic map itself may serve as a great source of information in its own right. Computations can be carried out on the topic domain without regard for the topic occurrences. Moreover, because of this separation, different topic maps can be overlaid on information pools to provide different views to different users [80]. Finally, topic maps created by different authors may be interchanged and even merged. This may even lead to a new business: an "information broker" can design topic maps and sell them to information providers, or link them to information resources and than sell them to end-users [83].

In [41], topic map authoring is examined. In particular, the issues of determining suitable topic, association and occurrence role types according to the nature of the application at hand, deciding the topic set of the scope, providing proper association types to obtain comprehensible navigation routes in the data set and automation of topic map creation process are discussed. In [83], the design and creation of a topic map is described as an incremental process, which includes declaring types of topics/associations/roles at the first step, and generation of instances of topics, associations and occurrences in the next step. For fairly large sets of information resources, the development of semi-automated or automated tools is necessary to make creation of topic maps scalable and implementable in practice.

Topic map templates is an issue mostly defended by [82]. A topic map template is itself a topic map consisting of all the constructs that have declarative meaning for another topic map. Furthermore, [82] defines the key issues that may contribute to a topic map schema. Topic map templates, type hierarchies,

association properties, inference rules and constraints are the key issues considered in this respect. Semantic validity of topic maps and constraint mechanism are further discussed in [64].

Publicly available topic map processors are provided in [16, 17] along with example topic maps authored in slightly different syntax. In [3, 13], commercial topic map authoring tools are presented. Topic-map embedded proceedings of GCA conferences are provided in [3] and their creation is discussed in [45]. Applying topic maps on top of hierarchical multi-user document repositories and alternative GUIs to ease the navigation and creation of topic maps are discussed in [37]. A commercial topic map search engine is presented in [10]. A reference implementation of topic maps is applied at a France portal site, *Quid*, which has more than 200,000 hyperlinks [3]. In [61], topic maps are used for semantic tagging of a corpus.

In [81], the similarity of topic maps and graph-based knowledge representation methods in the AI literature is discussed. Semantic networks, associative networks, partitioned networks and knowledge (conceptual) maps are some of the names of such models for representing knowledge structures in the field of AI. The study reported in [81] identifies the earliest work in this area as the existential graphs, invented by the philosopher Charles S. Peirce at the end of 19th century. One of the most developed schemes in this scope is the conceptual graphs, which is claimed to be as expressive as first order logic. In [81], authors suggest that topic maps add the *topic-occurrence* axis to the *topic-association* model, which is traditionally captured in the above AI models, and thus integrates the notion of knowledge representation with knowledge management.

Further discussion about the interchangeability of topic maps and semantic networks is provided in [63]. *SemanText* is a prototype system reported in this work, which builds semantic networks from the input topic maps. Note that, semantic network applications usually assign weights to the statements or facts captured in the network to denote the certainty value of assertions. Topic maps do not specify such an explicit mechanism to add weights to its constructs. However, the web information space (WIS) model described in this thesis attaches

"importance values" to the metadata entities of the model, as it is described in Chapter 3.

Finally in [69], a linear and a graphical notation are described for topic maps, inspired from the notation of semantic networks and conceptual graphs. This work further outlines a topic map query language based on the proposed linear notation and SQL.

2.2.2 XML Topic Maps (XTMs)

The essential motivation behind the XML Topic Maps effort is providing a more concrete representation of topic maps standard, which is originally described in terms of rather general SGML architectural forms [44]. As [75] identifies, XML facilitates the topic map creation process by allowing more readily incorporation of additional information resources and easier use of topic map constructs. On the other hand, topic maps may extend the power of XML in terms of making information self-describing, as they provide the opportunity of overlaying different perspectives on the same information pool. And using different maps allows different levels of separation of metadata and data at an information resource [75].

XML representation of the standard [44] is expected to encourage the use of a syntactic and functional subset of the topic maps in the mass market and to facilitate the applicability of the paradigm to the WWW. One open organization that is founded with similar concerns is "TopicMaps.org" [25]. The design goals of this research body includes developing an XTM standard which is conformant to XML and its related technologies (Xlink, Xpointer), usable over web and supports a wide range of applications, among others. The specification [32] published by this team includes a conceptual model, a syntax model along with an XTM DTD [33] and a processing model for XML Topic Maps [31].

The conceptual model is intended to give a basic understanding of XTM paradigm to an application programmer who will make use of the specified XTM

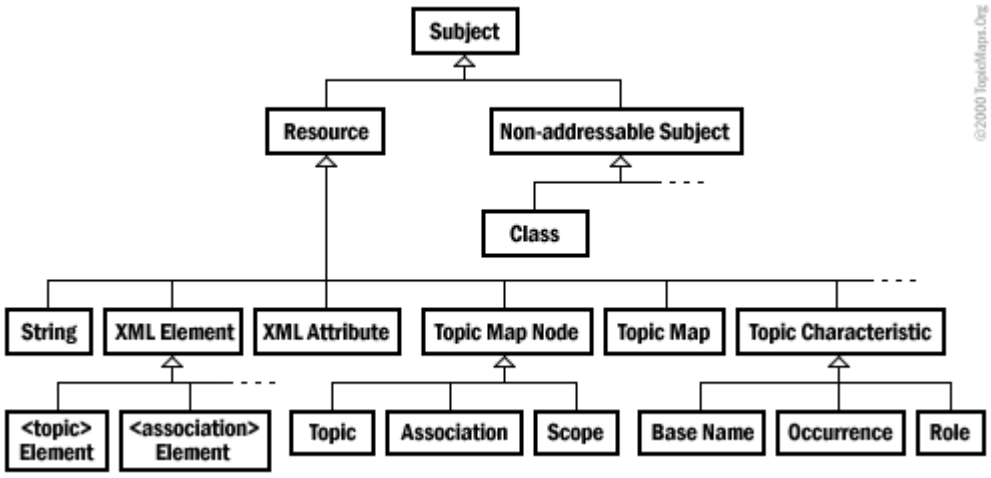

Figure 2.1: Class hierarchy for XTM conceptual model

interchange syntax (e.g., DTD [33]). The model is provided in [32] using the UML [19] notation. An overview of the class hierarchy is given in Figure 2.1.

The syntactic model specifies a set of XML tags to represent the (subset of) topic map constructs as we have described in the previous section. A brief discussion of the published XTM DTD is given in Appendix D. Note that the DTD simply facilitates the use of topic maps paradigm in the web environment by providing an XML interchange syntax. An example XTM document conformant to [33] is provided in Appendix E.

XTM documents are actually a form of representation for topic maps. To reconstitute the meaning of the information captured in a topic map structure, further processing is needed, rather than simply parsing the document [31]. Such a processing step couples certain operations with the mark-up in the DTD and thus yields an internal representation, called "topic-map graph", which is different from an ordinary DOM [29] tree. In accordance with [31], a topic map graph may consist of three basic types of nodes and the arcs connecting them. The elements <topic>, <association> and <scope> in an XTM document explicitly require the creation of corresponding node types for these elements. However, by the use of referencing mechanism in the XTMs (see Appendix E), the existence of a node in the graph may be required implicitly as well. Thus, it is possible that a topic map graph includes more nodes than its explicit *node-demanding* elements;

or it may have fewer nodes due to the merging of topic nodes that correspond to the same real world subject. The rules to create nodes and arcs from a given XTM document and the mechanisms to decide on the equality of two topics are described in [31], which was a working draft at the time of authoring this thesis. Merging of two topics in the same or different maps is achieved according to the *subject-based* and *name-based* merging constraints. The mechanism for inferring the identity (subject) of a topic in the subject-based approach is achieved by the use of a specific <subjectIdentity> element which points to a web resource that defines the topic unambiguously (see Appendix D). The name-based merging constraint is similar to the approach employed in this thesis used to detect the similarity of topic entities with respect to their names and domains (see Chapter 3).

In [75], the author identifies and argues a number of technical issues on the XTM specification, such as linking and addressing issues, including templates in the specification, specifying association properties and inference rules, and addressing the public topics. The issues mentioned above are also discussed at the XTM-WG mailing list [24] under a number of sub-groups, and the resulting document repository is publicly available in [26]. Two XTM conformant topic map engines are presented at [4, 17]. In [23], XSLT scripts to convert example ISO topic maps into XTM syntax is provided. Tools for XTM creation and navigation are reported at the sites [10, 8, 5]. At [18], an alternative XTM processing model is proposed. In [85], an object model is derived from the XTM specification and expressed in Object Definition Language to allow the use of Object Query Language for XTMs. The requirements for a topic map query language are further described in a number of working drafts [18]. Finally at [20], an exhaustive list of resources for topic maps and XTMs is provided.

2.2.3 RDF Standard

RDF (Resource Description Framework) is another specification to annotate resources with metadata in an interchangeable manner. The specification includes

two parts: the "Model and Syntax Specification" [49] and the "Schema Specification" [70]. In the following, we briefly summarize RDF and discuss its similarities to and differences from topic maps [62, 49]. In essence, RDF consists of an abstract data model, two XML-based syntax and an XML-based Schema language. The abstract model defines *resources*, their *properties* and *statements*. A resource may be anything that is being described using an RDF expression. It might be either accessible or not via the web (such as a web page, a web site or a printed book) but it is always identified by a URI [2]. A property is a specific characteristic or aspect used to describe a resource, and may be assigned to a structured entity or to another resource. A resource with a named property and its value contributes to a statement.

RDF is similar to topic maps effort in the sense that both specifications are intended to provide a mechanism for describing metadata and attaching it to information resources in an interchangeable manner by using SGML and/or XML syntax. However, as it is stated in [38, 79], RDF starts from the resources and annotates them, which may (optionally) extend to and end up at an abstract knowledge layer. In contrary, topic maps start with describing the topic domain and creates -roughly- a semantic network *above* the information resources, which may optionally be linked to the "occurrence domain". Thus, RDF is said to be suitable for *"resource-centric"* applications whereas topic maps apply to *"topic (knowledge)-centric"* applications.

In [74], convergence between the two proposals is discussed. Semantic Web [43] is an RDF schema-based effort to define an architecture for the web with a schema layer, logical layer, and a query language. A number of RDF-based works such as [50] is presented in the semantic web workshop [30].

2.3 Exploiting Metadata for Web Searching and Querying

In [56, 72], a metadata layer implementation, so-called *structured maps*, is presented. Structured maps are inspired by the Topic Maps, but employs a relational database schema to capture the metadata layer. The use of relational model also permits employing SQL to query the metadata. The work presented in this thesis differs from the latter approach in the following essential points (i) our web information space model is defined in the conceptual level, implying that the use of XML Topic Maps and relational databases in the prototype implementation is an arbitrary choice among other possibilities, (ii) the language developed for querying our model extends SQL remarkably, as querying the semi-structured data would exhibit differences in nature, such as approximate and ranked query evaluation. After all, we propose a much general framework with respect to [56] in this thesis.

WebML [88] is primarily designed for knowledge and resource discovery on the net. WebML is an SQL-like declarative language enriched with primitives that allow interactive querying with an OLAP (OnLine Analytical Processing)-like interaction (i.e., drill-down, roll-up, slice, dice, etc.). The language makes use of a Multi-Layered Database Model (MLDB) [66, 87] to provide layers of relational tables over the web resources (that happen to be the primitive layer) which are obtained by successive transformation and generalization of lower layers. The construction process assumes the presence or availability of descriptive metadata, either provided by document authors or extracted automatically. Besides, a set of concept hierarchies are employed in MLDB to facilitate the generalization of lower layers to higher ones. Although both WebML and SQL-TC exploits the metadata, the former language is highly focused on data-mining operations, whereas the latter is designed to provide a database-like view for querying the web-resources. In the web information space model of SQL-TC (see Chapter 3), there is a single layer of metadata stored in an expert advice repository, and concept hierarchies may be captured by *typing* the topics and using special metalink types (such as *SubTopicOf, SuperTopicOf*, etc.). Another distinguishing feature of SQL-TC is

allowing users to query in terms of the *relationships* of topics (effectively following the *metalinks*, in some sense), but not only the topics themselves.

The C-Web project [54] 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 knowledge engineers and domain experts, (ii) publishing information resources using the terminology of conceptual schema, (iii) enabling community members to query and retrieve the published information resources. To achieve the first goal, C-Web integrates *domain specific ontologies* and *hierarchical thesauri* -whenever available- to generate a *schema*, which is then exported in RDF syntax. While describing resources in schema terms, C-Web exploits any existing structure in the resources, and unstructured resources could be described manually through an editor. For instance, for a set of XML documents with a specified DTD, C-Web would define a mapping between its schema and the DTD, similar to our discussion in Chapter 3.4. Finally, querying facilities are provided by the language, so-called, RQL. Note that, the basic ideas and motivation of the C-Web project and this thesis are quite close, but the features of the proposed query languages and the aspects of the underlying metadata model, as well as the framework to express them differ (i.e., XTM vs. RDF).

While we propose a metadata-based search and querying for specific *subnets* on the web, the proposed web query languages in the literature have the broader goal of querying the web as a whole. One can view the world wide web as a directed graph and formulate queries based on the contents of web pages [60]. Query languages for the web include WebSQL [73], WebOQL [40], W3QL [68] and STRUQL [59]. There is also the issue of integrating information on different web sites (using "wrappers" [65]) and providing query capabilities over multiple web sites [71]. QUEST [42] system is designed to allow querying semantic structure of the web documents that are written in the markup language OHTML. Lorel [35] and UnSQL [52, 51] are among the query languages originally developed for semi-structured data querying. More recently, a number of query languages (such as XML-QL [57], XQL [84]) especially tailored for XML are proposed [46].

Chapter 3

Web Information Space Model

In this section, we present the topic-based web information space model as illustrated in Figure 3.1. 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. They may have bulk text in various formats (e.g., ascii, postscript, pdf, etc.), images with different formats (e.g., jpg), 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 represents an occurrence of a topic within an information resource. For example, the topic (with name) "Van Gogh" occurs multiple times as HTML documents within the documents of the information resource "Online Collections of the Smithsonian Institution" [9], and each such HTML document occurrence constitutes a topic source. 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 where the topic source starts within the

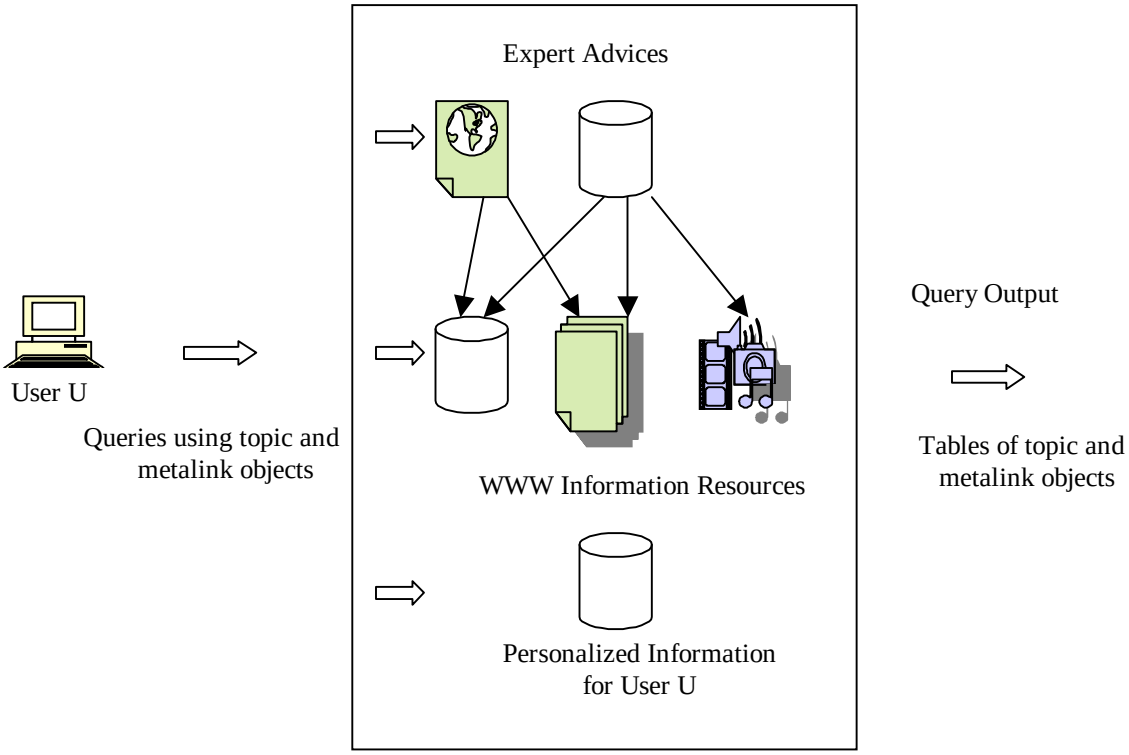

Figure 3.1: Web Information Space Model and Queries

document and where it ends, *LastUpdated, LastVisited, 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.).

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 each query explicitly, and we have n experts, E_i, $1 \leq i \leq n$. Each domain expert 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 [32]) in information resources. Example topic names are "database" (a keyword) and "Understanding the United Nation's Global Warming 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 the similarity of two topics on the basis of their names, we employ the 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 of "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", respectively. Therefore, for a web-based information resource on finance (e.g., Microsoft MoneyCentral [6]) a source for topic "stock market" can be at a beginner (i.e., detail level 1) level, denoted by StockMarket1 (e.g., only "Basic Investing" and "Stock Market Indexes"). Or, it may be at an advanced (i.e., detail level 3) level of StockMarket3 (e.g., "Risk Analysis and Random Walk Theory"), etc. 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 $i > j$. Also 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. Some of these additional attributes are discussed in the topic map standard and described in detail in [44].

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, $1 \leq i \leq n$, 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).

(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. The Don't-Care value is used when merging multiple expert advices.

Example 3.2.1 *Assume that the expert E assigns the following topic advice:*

TAdvice(E, TType="Diabetes", TName="*Diabetes Surgeries*", TDomain="New Patient Training") = 0.3

TAdvice(E, TName="Diabetes Management", TDomain="Patient Training") = 1
TAdvice(E, TName="*Kidney Complications*", TDomain="Nurse Training") = 0.7
TAdvice(E, TName="Advantages of Extreme Sports for Diabetes") = No
TAdvice(E, TName="Professional Sports and Diabetes") = Don't-Care

where * denotes a wildcard character that matches any string. The first 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. The second topic advice states that for patient training, the topic name "Diabetes Management" of any topic type is of highest importance. The fourth topic advice states that for any domain and any type, the topic "Advantages of Extreme Sports for Diabetes" is rejected. And, as the last topic advice, the expert does not care about the topic "Professional Sports and Diabetes" and does not object to its use as a topic name by other experts.

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).

Example 3.2.2 *Using the CWA-Don't-Care assumption and TAdvice() specifications in Example 3.2.1, the expert specifies:*

TAdvice(E, TName="Drinking", TDomain="Professional Sports") = "Don't-Care".

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 document that contains the topic source.

- *Start-Marker* (address) indicating the starting address of the topic source relative to the beginning of the document. For topic sources in XML-based web documents, this attribute is redundant.
- *End-marker* (address) indicating the end address of the topic source relative to the beginning of the document. For topic sources in XML-based web documents, this attribute is redundant.
- *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,* etc.

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

Example 3.2.3 *Assume that the expert E assigns the following topic source advice:*

SAdvice(E, TType="disease", TName="Cancer", TDomain="PatientCare", Web-Address=www.mayo.org) = 0.5

SAdvice(E, TType="chronic disease", TName="Diabetes Management", TDomain="PatientCare", Web-Address=www.ada.org) = 1

SAdvice(E, TName="Breaking Diabetes News", TDomain="Diabetes Research", Last-Modified=(Now - 2years)) = No

The last source advice states that, for the topic name "Breaking Diabetes News" (of any topic type) in the topic domain "Diabetes Research", any source that was last modified two years ago is rejected.

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

Topic Metalinks represent relationships among topics. Metalinks may have as attributes types, roles, domains, etc. An example metalink type is *RelatedTo*. For example, the topic with name "Downhill Skiing" at level 1 is *RelatedTo* the topic with name "Super GS Ski Racing" at level 1, represented using the notation "Super GS Ski Racing^1 $\rightarrow$*RelatedTo* Downhill Skiing^1". The notation $\rightarrow$ *RelatedTo* represents an instance of the metalink type *RelatedTo*. As another example, consider the learning-related metalink type *Prerequisite* and the metalink instance "Diabetes Complications^2 $\rightarrow$ *Prerequisite* Diabetes^1 " stating that "Understanding of the topic Diabetes at level 1 (or higher) is the prerequisite to understanding/learning the topic Diabetes Complications at level 2". Within the context of electronic books [78], a sound and complete set of axioms for the *Prerequisite* relationship has been given. Similar to prerequisite metalinks, one may use the notion of topic *corequisites*, specifying the corequisites of the given topic. Yet another metalink relationship can be the *LeadsTo* relationship that states, for example, that "the topic relational model *LeadsTo* the topic query languages". Thus any relationship involving topics deemed suitable by an expert in the field can be a topic metalink.

SubTopicOf and *SuperTopicOf* metalink types together represent a topic composition hierarchy. As an example, the topic "database" is a super-topic (composed) of topics "data model", "query languages", "query processing", etc. And the topic "relational algebra" is a sub-topic of "query languages" and "relational model".

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, $1 \leq i \leq n$, 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-Care}. E_i.Metalinks denote the set of metalink types defined by the expert E_i. Similarly, E_i.Topics denote the set of topics defined by the expert E_i.

Example 3.2.4 *Assume that the expert E states the following metalink signatures:*

E.Metalinks={*RelatedTo* : topic $\rightarrow$ topic, *Prerequisite* : SetOf topic $\rightarrow$ 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. Now, assume that the expert E states the following metalink (instance) advice:

MAdvice (E, Diabetes Care^1 $\rightarrow$ *RelatedTo* Diabetes Complications^1)=0.8
MAdvice (E, Diabetes Care^1 $\rightarrow$ *Prerequisite* Healthy Eating^1)=1
MAdvice (E, Diabetes Surgeries^3 $\rightarrow$ *Prerequisite* Diabetes Care^1)=No

The first metalink states that the importance value of the metalink "the topic Diabetes Complications at the beginner level (1) is related to Diabetes Care at the beginner level" is reasonably high (0.8) (There may be other causes for diabetes complications). The second metalink states that understanding healthy eating at a beginner level is a prerequisite to understanding diabetes care at a beginner level. And, the last metalink states that understanding diabetes care at beginner level or above is not a prerequisite to understanding the topic diabetes surgeries at an expert level.

We assume that there are multiple experts (and, thus multiple expert advices) on information resources, with each expert specifying (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 the possible conflicts among them. An example illustrating this situation along with a user preference-based solution attempt is provided in the Example Query 4.2.3 of Chapter 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. The prototype system described in Chapter 5 is developed using XTM documents as expert advice repositories. Note that XTM is a recently initiated effort, and there are many issues with XTM to be resolved, conceptually or practically, such as namespaces, topic map merging and processing model [31] implementations. In this sense, our research also contributes to resolve some of these issues as it provides a practical application for the use of XTMs.

3.2.3 Topic and Metalink Closures

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 to return query 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.

Assume that the expert E specifies four topic entities A (with name "Diabetes Care"), B (with name "Insulin Shot Plan"), C ("Diabetic Food Plan"), and D ("Carbohydrate Counting"), and the metalinks A$\rightarrow$ *Prerequisite*(B, C), and C$\rightarrow$*Prerequisite* D. For the sake of simplicity, assume that all detail level values are 1, and ignored. And, the user U asks for topic sources of topic A, subject to the advice of expert E. Since the *Prerequisite* metalink is transitive, the user's

request about topic A needs to be expanded by a *topic closure* of A with respect to the *Prerequisite* metalink instances specified by E. Also note that we can always decompose the right-hand side (RHS) of a *Prerequisite* metalink. That is, the metalink A→*Prerequisite*(B, C) is equivalent to the metalinks A→*Prerequisite* B and A→*Prerequisite* C.

Thus, the correct response should include the topics A, B, C, D, and their resources. In this section, we briefly discuss how to compute *Topic Closures*, i.e., given a set X of topics, obtaining the closure X^+ (the set of topics that are also logically implied and thus are in the response). 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.

Consider the *Prerequisite* metalink type. In [78], the following set of sound and complete axiomatization for the Prerequisite metalink type is given.

Case 1. Prerequisite metalinks are not left-hand-side (LHS) decomposable (that is, A, B→*Prerequisite* C is not equivalent to the metalink A→*Prerequisite* C and the metalink B→*prerequisite* C), and are allowed to be cyclic.
Axioms: Let X, Y, and Z denote sets of topics.

- Subset-Reflexivity. If Y ⊆ X, then X→*Prerequisite* Y
- Augmentation. If X→*Prerequisite Y*, then XZ →*Prerequisite* YZ for any Z
- Transitivity. If X→*Prerequisite* Y and Y→*Prerequisite* Z, then X→*Prerequisite* Z

These are the so-called Armstrong's axioms [86].
Case 2. Prerequisite metalinks are not LHS-decomposable and are acyclic.
Axioms: Let X, Y, Z and W denote sets of topics.

- Pseudo-transitivity. If X →*Prerequisite* Y and WY →*Prerequisite* Z, then WX →*Prerequisite* Z

Input: Set of topics X, Set M of metalink types:
Var X: SetOf topic;
M: SetOf metalink;
Change: Boolean;
OldX : SetOf topic;
Output: Closure set X^+ of topics X with respect to metalink types M;
Var X^+ : SetOf topic;
begin
X+ := X;
OldX := X^+;
Change := True;
while (Change) do
$X^+ := X^+ \cup \{y\}$ such that there is a metalink instance of type T
in M of the form Z $\rightarrow$T {y} where $Z \subseteq X^+$ and $y \notin X^+$;
if (X^+= OldX)
then Change := False
else OldX := X^+;
endwhile
end

Figure 3.2: Topic closure algorithm (involving multiple metalink types)

- Split/join. If X$\rightarrow$*Prerequisite* YZ, then X$\rightarrow$*Prerequisite* Y and X$\rightarrow$*Prerequisite* Z, and vice-versa

In [78], it is proven that these axioms are sound and complete.

Case 3. Prerequisite metalinks are LHS-decomposable.

We first decompose the LHS of all metalinks so that all metalinks have a single topic in the left and the right hand sides. And, then the only axiom is

- Transitivity. If A$\rightarrow$*Prerequisite* B and B$\rightarrow$*Prerequisite* C, then A$\rightarrow$*Prerequisite* C where A, B, and C are topics.

Note that in all three cases, the topic closure X^+ of a set X of topics can be found by using an $O(n.l)$ topic closure algorithm, as specified in Figure 3.2, where

n is the number of prerequisite metalinks and l is the length of the encoding for a prerequisite metalink [78].

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$\rightarrow$*RelatedTo* B, then B $\rightarrow$*RelatedTo* A
- Transitivity. If A$\rightarrow$*RelatedTo* B and B$\rightarrow$*RelatedTo* C, then A$\rightarrow$*RelatedTo* C

Prerequisite Axioms: Armstrong's axioms (Case 1 above).
RelatedTo and *Prerequisite* mixed axioms:

- If X$\rightarrow$*Prerequisite* A and A$\rightarrow$*RelatedTo* B, then C$\rightarrow$*RelatedTo* B for all C where C $\in$ X.
- If X$\rightarrow$*RelatedTo* A and A$\rightarrow$*Prerequisite* B, then C$\rightarrow$*RelatedTo* B for all C where C $\in$ X.

With these axioms, we can find the topic closure X^+ of a set X of topics by using the $O(n.l)$ closure algorithm in Figure 3.2, where n is the number of *Prerequisite* and *RelatedTo* metalinks, and l is the max length of the encoding for a *Prerequisite* or a *RelatedTo* metalink.

3.3 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.

3.3.1 User Preferences

In this thesis, we employ user preference specifications along the lines of Agrawal and Wimmers [36]. The user U specifies his/her preferences as an ordered set of Accept-Expert, Accept-Expert-Metalink-Importance-Threshold, etc. statements, as listed below.

(Accept-)Expert (U)= $< E_i$,..., $E_j > 1 \leq i, j \leq n$ (sequence of experts whose advice is to be satisfied)

(Accept-Expert-)T(opic-)Imp(ortance-Threshold) (U) = {(E_i, ValueThreshold),..., (E_j, ValueThreshold)} (set of topic importance value thresholds, one for each expert)

(Accept -Expert-)M(etalink-)Im(portance-Threshold) (U) = { (E_i, ValueThreshold),..., (E_j, ValueThreshold)} (set of metalink importance value thresholds, one for each expert)

(Accept -Expert-)S(ource-)Imp(ortance-Threshold) (U) = {(E_i, ValueThreshold),..., (E_j, ValueThreshold)} (set of source importance value thresholds, one for each expert)

Reject-T(opic-Attribute) (U) = {Attribute=value|$< E_i$, Attribute=value>}(set of topic attribute values to be rejected)

Reject-S(ource-Attribute) (U) = {Attribute=value|$< E_i$, Attribute=value>} (set of source attribute values to be rejected)

(Expert-)Conflict-R(esolution) (U) = Ordered-Accept | Accept-All | Manual
(Accept advices in an ordered manner, as listed by the Expert() function above; or accept all advice, ignoring all conflicting advice; or let user choose which advice to accept in the case of a conflict)

We illustrate these preference functions with an example.

Example 3.3.1 *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-T (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-S (John-Doe)={Web-Address=www.dirtypolitics.com}

Conflict-R=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)

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 (and less than the maximum detail level attribute 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 3.3.2 *Assume that the user John-Doe knows topics with names "Racquetball" at an expert (3) level and "Tennis" at a beginner (1) level, specified as*

UKnowledge(John-Doe) = {(TName="Racquetball", 3), (TName="Tennis", 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. 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, i.e., the user's knowledge level about all topics is zero.

Here in this thesis, we assume that the user profiles are XML documents with single/multiple user profiles.

3.4 A Discussion on the Applicability and Practicality of WIS Model

Real-world practicality of the Web Information Space (WIS) model described above is an important question to be considered to justify the work represented in this thesis. Clearly, among the three components of the WIS model, the *information resources* are already conformant to the current state of the web, and the data about *user knowledge/preference* can be gathered and managed employing techniques similar to those described in many other works (see [55] as an example). The remaining component, *expert advice*, is the heart of our system distinguishing it from other works, and thus it is useful here to speculate about creation and/or maintenance of such metadata repositories, even though it is not the major issue focused in this thesis.

As we have mentioned in the Introduction chapter, the approach proposed in our work does not address the whole *web* to solve the problem of information retrieval, but we rather concentrate on 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 and query techniques proposed later in this study. A family of subnets that we can consider is exemplified in the previous sections: a collection of web sites that belong to a particular domain such as a company, organization etc. (e.g., ACM SIGMOD sites or Online Collections of Smithsonian Institution). Though, 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, as it would be discussed in this section.

Let us begin with considering the case in which a metadata repository is associated with a collection of sites/documents belonging to a certain organization or domain. In the most straightforward scenario that may be assumed, the metadata entities are designed by domain experts(s) from scratch and stored either manually or semi-automatically, possibly by using a topic map editor, such as the ones at [3, 10]. Clearly, the creation and maintenance is an expensive work that

```
// book.dtd
<!ELEMENT (book *)>
<!ELEMENT book(title, author, publisher, date, cover artist, ISDN)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT publisher (#PCDATA)>
<!ELEMENT date (#PCDATA)>
...
```

Figure 3.3: Book.dtd

```
// review.dtd
<!ELEMENT (review *)>
<!ELEMENT review(artwork, reviewer, rate, notes)>
<!ELEMENT artwork (title, author, publisher)>
<!ELEMENT reviewer (name, surname, occupation)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT publisher (#PCDATA)>
<!ELEMENT rate (#PCDATA)>
<!ELEMENT notes (#PCDATA)>
...
```

Figure 3.4: Review.dtd

requires vast amount of human work. However, it would be compensated by the stability of metadata and the opportunity of locating any piece of data in a fast and exact manner. Additionally, we envision that there may exist much more efficient approaches to create expert advice repositories, making use of the advent of XML over the web. Besides, such approaches may allow to deal with information resources that do not necessarily "close" in terms of physical placement or the organizations that own the data.

To illustrate this latter provision, consider the following scenario. As the number of web-documents published in XML increase very rapidly, just like the protocols to transmit them [14] and applications to process them, the book publishers decide to adapt a set of DTDs (such as [11]) to take their part in the e-business market. Similarly, a number of journals that provide book reviews adapt a set of DTDs to catch up with the trend. Assume that the Book and Review DTDs provided in Figure 3.3 and 3.4 are two such DTDs that are specified

M = { *//DTD set employed in the mappings*
nms1: "http://publishersassociaiton.org/book.dtd",
nms2: "http://weeklyreviewers.org/review.dtd"
// The use of element or attribute names below would also include DTDs' own namespaces in case of the ambiguity.

// Topic generation rules
t1: <nms1, Tname = ValueOf (books.book.title),
TType=EL-NAME(book)>,
t2: <nms1, Tname = ValueOf (books.book.author),
TType = EL-NAME(author)>,
t3: <nms1, Tname = ValueOf (books.book.publisher),
TType = EL-NAME(publisher)>,
t4: <nms2, Tname = ValueOf (reviews.review.artwork.title),
TType = EL-NAME(book)>
t5: <nms2, Tname = ValueOf(reviews.review.reviewer.name +
reviews.review.reviewer.surname, Ttype = EL-NAME(reviewer)>,
t6: < nms2, Tname = ValueOf (reviews.review.rate),
TType = review rate)>,

// Metalink generation rules
m1: <*WrittenBy*: t2 → t1, Doc-id(t2)=Doc-id(t1)>,
m2: <*ReviewedBy*: t5 → t, where t ⊆ Merge(t1, t4) or t ⊆ t4,
Doc-id(t5)⊆ Doc-id(t)>,
m3: <*RatedAs*: t6 → t1, Doc-id(t6)=Doc-id(t1)> }

Figure 3.5: Mapping *M*

in the DTD sets by the above parties respectively.

Given these DTDs, an expert may designate the values of a number of elements (or attributes) to be topics, and further define particular relationships among these topics. Then, say, a *robot* -similar to those employed in today's search engines- could traverse the web and create metadata entries with respect to the mapping of the domain expert for each document conformant to the given DTDs all over the web. Assume that a domain expert specifies the *mapping M* of Figure 3.5 between the above DTDs and our expert advice model entities. Note that, a similar mapping-based approach is also outlined in [54].

In this mapping, the first two lines specify that the element and attribute names provided would be in the namespaces [1] of the respective DTDs. Next, *generic topics* are defined according to these DTDs. For instance, the line for *t1* forces the web-robot to create a topic whose TName is the string value which is obtained by following the path books.book.title in any XML document conformant to the first DTD in our example. The TType of the topic is designated as "book", and the notation EL-NAME(book) just declares that it is the name of an element in the given namespace. Note that, the domain expert could set this value to an arbitrary string, say "novel", if (s)he wishes to do so (such as the TType "review rate" for *t6* above). The value of a topic may be not only the value of a single element or attribute, but even the concatenation of both. For instance, the topic generation rule *t5* enforces that the TName of the created topic will be the concatenation of the values of two elements, <name> and <surname>, from the conforming documents.

To define the metalinks signatures, the domain expert declares the metalink type explicitly and the set(s) of generic topics for which this particular relationship holds. For instance, the rule for generic metalink *m1* indicates that there exist a *WrittenBy* relationship between (generic) topics *t2* and *t1* that are extracted from the the same document. In the metalink signature *m2*, the function Merge() is explicitly called to force the web robot to guarantee that instances of two generic topics (*t1* and *t4*) from different documents must be merged if they refer to the same real world entity. The issue of merging topics within a topic map is further discussed in [31], and ideally, an XTM corresponding to an expert advice repository should not include more than one topic that refer to the same real world entity. Thus, the detection of duplicates and merging them is an essential responsibility of an XTM processor. In the above mapping, the use of Merge() function by the expert is intended to provide further clue to the processor system. The detailed design and interpretation of such mapping rules is out of the scope of this work and may be further investigated on its own right.

A domain expert may provide such a mapping simply by interacting with a

[1] Please note that, the use of word "namespace" here is different from the *XML-namespace* as it is known in the literature.

```
<!DOCTYPE books SYSTEM
"http://publishersassociaiton.org/book.dtd"> <books>
   <book>
      <title> A girl who loved Tom Gordon </title>
      <author> Stephen King </author>
      <publisher> Scribner </publisher>
      ...
   </book>
   ...
</books>
```

Figure 3.6: Example document conforming to Book.dtd

```
<!DOCTYPE reviews SYSTEM "http://weeklyreviewers.org/review.dtd">
<reviews>
  <review>
     <artwork>
        <title> A girl who loved Tom Gordon </title>
        <author> Stephen King </author>
        ...
     </artwork>
     <reviewer>
        <name> Michael </name>
        <surname> Ray </surname>
       ...
     </reviewer>
     <rate> 3 / 5 </rate>
    <notes> This book is about  a... </notes>
   ...
  </review>
  ...
</reviews>
```

Figure 3.7: Example document conforming to Review.dtd

GUI-based tool or by specifying a set of rules or in any other convenient manner. Given such a set of rules to our web-robot, let us assume that the example documents in Figures 3.6 and 3.7, which are conformant to above DTDs, are encountered at the web sites www.amazon.com and www.bookreviews.com, respectively.

Tid	**TDet.**	**TType**	**Tname**	**TDom**	**TAdv.**	**Source**
T1	-	author	"Stephen King"	-	-	{S1, S2}
T2	-	book	"The girl who loved Tom Gordon"	-	-	{S1, S2}
T3	-	publisher	"Scribner"	-	-	{S1}
T4	-	reviewer	"Michael Ray"	-	-	{S2}
T5	-	review rate	"3/5"	-	-	{S2}

Table 3.1: Extracted topics

Mid	**MType**	**MDom**	**Antecedent players**	**Consequent players**	**MAdv**
M1	WrittenBy	-	T1	T2	-
M2	ReviewedBy	-	T4	T2	-
M3	RatedAs	-	T5	T2	-

Table 3.2: Extracted metalinks

Sid	**Web-address**	**Role**	**Media type**	**LastUpdated**	**Detail level**	**SAdv**
S1	www.amazon.com	-	text	12.07.2001	-	-
S2	www.bookreviews.com	-	text	13.07.2001	-	-

Table 3.3: Extracted sources

In Tables 3.1 to 3.3, the metadata entities are created by applying the rules given in the mapping M to these example documents. For the sake of saving place, the resulting XTM document is shown in a tabular manner. The table entries are in conformance with the example expert advice repositories provided in Appendix A. Note that, the mapping could also provide bindings for TDomain and importance values in a similar manner, but it is not considered here since

this section is solely intended to give the flavor of ideas for creating expert advice repositories.

Note that the above approach gives way to much more precise querying opportunities than an ordinary XML document querying approach. Any query in the current XML query languages should specify particular XML documents as sources, whereas the metadata we have provided covers a distributed set of documents over the web, along with almost providing the querying power of a central database system. For instance, using the above expert advice repository, one can pose the query "find all resources for the books written by Stephen King and rated 3 or more (in the average) by the reviewers", 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. Querying web using the WIS model is essentially discussed in the next chapter of this thesis.

Additionally, the robots may be constructed such that they may look for many different mappings during a single *crawling* session, and thus multiple metadata repositories may grow for different domains of interest. Here, the expert advice repositories are created in a semi-automatical manner, and once a mapping between a set of DTDs and the expert advice model is provided by the domain expert, the human intervention would be required at minimum, or may not be needed at all. Note that, the resulting metadata repository may refer to many different documents that can belong to different organizations that are both logically and physically located far apart from each other.

Yet another (somewhat less-likely) scenario for creating expert advice repositories may benefit from the metadata which is actually *embedded* in the information resources. For the time being, expert advice databases are considered to be separate metadata repositories including topics, their metalinks, the links to their actual occurrences -whenever possible- and the importance values associated with them. In this sense, an expert database reflects the perception of concepts in an information pool from the perspective of its authors, the domain experts.

On the other hand, we envision that, just like any other author, the original author of an information resource himself/herself may also appreciate to reflect

his/her attitude in terms of the expert advice model entities. For instance, the webmaster of an internet site would actually have some idea on what topics the site is about and what kind of metalinks should exist among them. In this case, the webmaster can incorporate the XTM constructs directly to the resource document, assuming that the document is prepared in XML and XTM documents are used to represent the expert advice repositories . This is similar to the today's -somewhat limited- practice of adding <meta> tags to an HTML document to specify metadata about the document. Besides, whenever the site is modified, the metadata can also be easily updated to express the current relationships. We identify some potential implications of such an approach as follows:

- Resource owners may submit their documents (which are now enriched by XTM constructs) to *XTM robots*, which would simply extract the metadata from the information resources. Such robots may create expert advice repositories in automated or semi-automated manners,
- Applications operating on these information resources (such as our query engine) may make use of the metadata at resources as another (somewhat distributed) expert advice imposed on the resource set.

Although we note the above issues, we will not focus on them, simply because for the time being, there is not much evidence that ordinary people or webmasters would be tended to include metadata in the information resources themselves, and even if they would, the reliability of embedded metadata is an open question. Thus, the work presented in this thesis relies on expert advice repositories that have been created by domain experts.

The scenarios described above just illustrate that with the advent of XML in the web environment, the burden of creating metadata may be considerably reduced, and can be left to robot-like agents. Thus, the term *subnet* in our framework may refer to a large and diversified collections of information resources associated with metadata, and locating the exact piece of required information in such collections would still be a non-trivial question, which we attempt to solve in the forth-coming chapters.

Chapter 4

Topic-Centric Query Language: SQL-TC

4.1 Overview and Basic Features

SQL-TC is an integrated SQL-like Topic-Centric language for querying web-based information resources, expert advice repositories and personalized user information. In this section, we outline the syntax of the language and its special constructs. Then, we illustrate the features of our language with queries over information resources, incorporating multiple expert advice and personalized user information. The queries demonstrate the notion of closure computation on particular metalinks and search of information resources. Next we provide a few queries that are posed solely on the expert advice information in order to illustrate additional features of SQL-TC.

Note that the query language presented here is a multi-database (multiple expert advice and user profiles, and multiple information resources) query language with heterogeneous data access. SQL-TC is strongly typed with each variable having a well-defined type. Furthermore, the language is general enough to be operational on any underlying expert data model, as long as the model supports metadata objects and their attributes.

First, we specify the syntactic constructs of SQL-TC. The formal syntax in extended Backus-Naur format is given in Appendix C.

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

In the **Stop after** clause, n denotes an integer variable to restrict the number of result tuples to be returned and m is a threshold value in the range [0, 1] to eliminate the tuples with importance values below this threshold. Note that the **Stop after** clause is adapted from [53].

4.2 Querying Web-based Information Resources

In this section, we illustrate SQL-TC by example queries. We assume that we have two experts whose advices are at www.sql-tc.com/king.xtm (expert E1) and www.horror-books.com/books.xtm (expert E2), respectively. The information resources are at http://www.stephenkinglibrary.com and http://www.stephen-king.net. As expert advice and user profile information, we use the instances provided in Appendices A and B, respectively. For example, from Appendix B, the user U prefers to accept first the advice of expert E1, and then, if there are no conflicts, the advice of expert E2.

Example 4.2.1 *(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 novels' detail level 4 reviews from the two information resources.

select [$topic.name, $sourceRef.web-address] **as T**
from resources http://www.stephenkinglibrary.com,
http://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", horror novelist, literature, E))
and $sourceRef = SourceOf($topic, 4, E) **and**
"review" **in** $sourceRef.roles
order by $topic **importance**
stop after 2 **most important**

This query returns novel names (i.e., topic names) and the web addresses of their reviews (i.e., topic sources) from two information resources on the web. The result of the query is a 2-column table. In the query, variables are prefixed by the $ symbol, constants are in quotes, and metalinks are in italics. 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 $\rightarrow$ novel[1]

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 "horror novelist" and TDomain of "literature", and specified by 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

[1]Recall that, along the lines of earlier work [78], the metalink instances are read in inverse order, from right to left.

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.

Please note that this query does not employ the user profile. Thus, 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 importance value of 0.8) are returned as shown in Table 4.1.

TName	**SourceRef.Web-address**
"Carrie"	http://www.critics.com/carrie.html
"The Stand"	http://www.critics.com/stand.html

Table 4.1: Output of the SQL-TC query in Example 4.2.1.

Example 4.2.2 *(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 http://www.stephenkinglibrary.com,
http://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", horror novelist, literature, E)
and *RelatedTo**("Wizard and The Glass", , literature, E)) **and**
$sourceRef = SourceOf($topic, 4, E) **and**
"review" **in** $sourceRef.roles
$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): topic→ topic

Note that *RelatedTo()* metalink in this query uses the topic variable $topic, which in turn has all four key attributes (i.e., TName, Ttype, TDomain, TAuthor) specified. In this query the user asks for the highest-valued tuple, not the highest-valued novel. Derived importance value computation of output tuples takes place, and the tuple in Table 4.2 is chosen.

Let us discuss the interpretation of this query using the expert repository and user profile instances in the Appendix. 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. The query result is given in Table 4.2.

TName	**SourceRef.Web-address**
"The Wasteful Lands"	http://www.critics.com/dark3.html

Table 4.2: Output of the SQL-TC query in Example 4.2.2.

Example 4.2.3 *(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", 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 http://www.stephenkinglibrary.com,
http://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**
$sourceRef= SourceOf($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 a main character of the novel "The Stand" 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), the novel "The Stand" has the main character "Jack Park", who 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 as this query employs more than one expert advice, the issue of possible 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, "Jack Park"→*InfluencedBy* "John Smith") = 0.8
MAdvice (E2, "Jack Park"→*InfluencedBy* "John Smith") = "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" [2] 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.

Example 4.2.4 *(Multilevel nested queries and the aggregation operators) Find the advice, at any level, of the expert www.sql-tc.com/king.xtm for a novel written*

[2]The threshold value Don't Care indicates that all metalink instances are accepted from this particular expert, regardless of their importance value.

by Stephen King such that the novel has the highest importance value according to the reviews at www.king-review.com.

```
select [$sourceRef.web-address] as T
from resources http://www.stephenkinglibrary.com,
               http://www.stephen-king.net
using experts www.sql-tc.com/king.xtm as E1
where $topic in E1.Topics and
      $sourceRef=SourceOf($topic,,E1) and
      $topic.TName in
            select [$topic1.TName]
            using experts www.king-review.com as E2
            where WrittenBy in E2.Metalinks and
                  $topic1 in E2.Topics and
                  $topic1=any (WrittenBy
                  (SimTName("Stephen King", horror
                              novelist, literature, E2))) and
                  TAdviceMatch (E2, $topic1) =
                        select max (TAdviceMatch (E2, $topic2))
                        where $topic2 in E2.Topics
```

The function TAdviceMatch (expert, topic) returns the expert's importance value "matched" to topic. If no matching takes place, the value of 0 is returned.

In the above query, three levels of select-subquery nesting are employed. The innermost query finds the highest topic importance value for any topic, assigned by the expert www.king-review.com by using the aggregate operator max in a way similar to SQL. The second-level select subquery returns the names of Stephen King novels that are assigned the maximum importance value according to the expert www.king-review.com. And the outermost query finds all sources, at any level, for the selected novels, advised by the expert www.sql-tc.com/king.xtm.

Example 4.2.5 *(Text search in the source) Find all sources, at any detail level, that are about Stephen King and contain the string "accident".*

select [$sourceRef.web-address] **as** T
from resources http://www.stephenkinglibrary.com,
http://www.stephen-king.net
using experts www.sql-tc.com/king.xtm **as** E
where $topic **in** E.Topics **and**
$topic.TName = "Stephen King" **and**
$sourceRef = SourceOf($topic,,E) **and** Contains($sourceRef, "accident")

Assume that the information resource located at www.newsweek.com/17-10-2000/tragic-accident.html mentions the topic "Stephen King" and contains the string "accident". In this case, the result of the query would be as in the Table 4.3.

SourceRef.Web-address
"www.newsweek.com/17-10-2000/tragic-accident.html"

Table 4.3: Output of the query in Example 4.2.5.

4.3 Querying Expert Advice Repositories

In this section, we provide two examples that solely query the expert advice repositories.

Example 4.3.1 *(Multiple Metalink Closures) Find all topic names of any type, which are related to the novels based upon Stephen King novels.*

select [$topic.TName] **as** T

using experts www.sql-tc.com/king.xtm **as** E
where *WrittenBy, RelatedTo, BasedUpon* **in** E.Metalinks **and**
$topic1 = **any** (*WrittenBy* ("Stephen King",, literature, E)) **and**
$topic = **any** ((*RelatedTo* (*BasedUpon*))*
($topic1.TName, $topic1.TType, $topic1.TDomain, E))

For this query, we assume the following metalink signatures.

WrittenBy (E): SetOf author→novel
RelatedTo (E): topic→topic
BasedUpon (E): SetOf base-topic→result-topic

Note that novels are topics, and the *RelatedTo* and *BasedUpon* metalinks return novels, movies, musicals, and persons. In the above query, the result is a table with rows that contain topic names for each topic entity qualifying the conditions. As in the previous examples, the notation *(RelatedTo(BasedUpon))** enforces topic closure computation. The algorithm in Figure 3.2 represents the interpretation semantics for this case.

We now briefly discuss the query evaluation using the instances in the Appendix. According to the expert advice at www.sql-tc.com/king.xtm (i.e., expert E1 in Appendix A), the set of novels "Carrie", "The Stand", "Wizard and The Glass", "The Wasteful Lands", "Drawings of Three", "Dark Tower" is written by Stephen King. The input to the multi-closure algorithm is these topics and the metalink set M that contains metalinks *BasedUpon* and *RelatedTo*. At the first iteration of multi-closure computation, X^+ is augmented by the movie "Story of Carrie" which is *BasedUpon* the corresponding novel "Carrie". Then the closure set is further expanded by "John Carpenter", which is *RelatedTo* this movie. There is no contribution of other novels to the closure set. At the second iteration of the while loop, the algorithm first retrieves the topics that are based upon the current set, which has been extended by "Story of Carrie", "John Carpenter" in the previous step. As "Carrie: The Musical" is based upon the movie "Story of Carrie", it is added to X^+, and no further topic is based upon the topics in this set. Then, the topics that are related to "Carrie: The Musical" are added to the

set, and "Broadway Season 1980" is the only such topic to be added. Again, no other topics are added to the closure set, as there is no topic that is *RelatedTo* the topics in the current set. The algorithm stops when the closure does not change in the next iteration. The result of the query is given in Table 4.4.

TName
"Carrie"
"The Stand"
"Wizard and the Glass"
"Wasteful Lands"
"Drawings of Three"
"Dark Tower"
"Story of Carrie"
"John Carpenter"
"Carrie: The Musical"
"Broadway Season 1980"

Table 4.4: Output of the query in Example 4.3.1.

Example 4.3.2 *(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] **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.

Mid	**MType**	**MDomain**	**Antecedent players**	**Consequent players**	**M-Adv**
M1	WrittenBy	{literature, horror}	[Stephen King, horror novelist, literature, -]	[Carrie, novel, literature, -]	1

Table 4.5: Output of the query in Example 4.3.2.

Using the expert advice in Appendix A, the query yields the output specified in Table 4.5.

Chapter 5

Implementation Issues

5.1 System architecture

In Figure 5.1, we provide the architecture of our system in its most general form. In what follows, we briefly explain the key components of this system. Implementation-specific parts are described in the next section.

- *Visual query interface* is a graphical user interface provided to users. Although the examples illustrated up to this point are textual, the implementation, when finished, will provide the user a GUI to ease the use of SQL-TC in the web environment.
- *Expert advice access interface* provides a uniform interface (schema) for accessing expert advice repositories with many possible kinds of underlying models or implementations.
- *User profile database access interface* is similar to the above one, providing a uniform interface to access underlying heterogeneous databases.

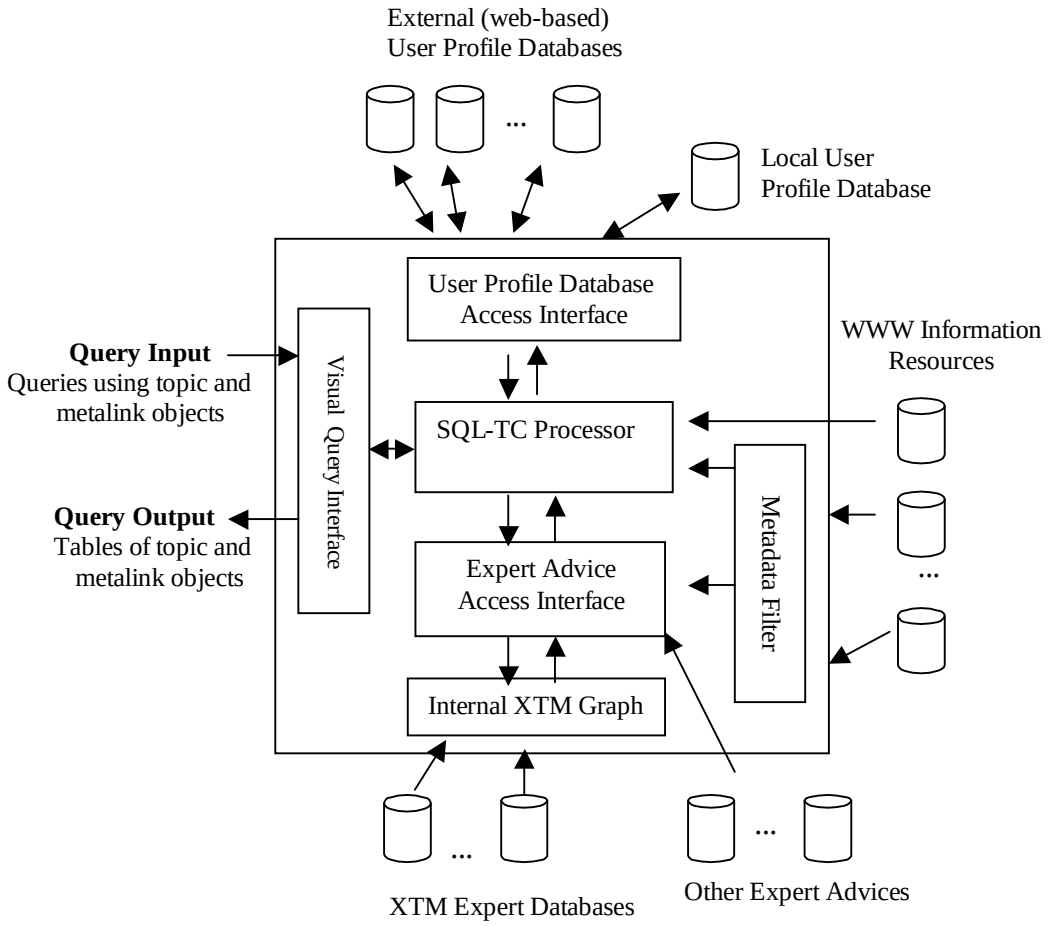

Figure 5.1: SQL-TC System Architecture

5.2 Prototype implementation

We have implemented a prototype system on the PC-Windows platform using Java 2 SDK^{TM}, Sun Java servlet engine and a web server provided within JSWDK (JavaServer Web Development Kit). In the prototype system, XTMs serve as expert advice repositories. All (virtual) information resources referenced from these repositories are assumed to be XML documents. In the next two sections, we discuss XTM-specific methods and tools to make use of them as expert advice repositories. Then, we outline a post-processing stage to store XTM data into expert advice object model. Finally, a visual interface that currently supports a few query templates is given. The user profiles have not been implemented in the prototype system yet, but Appendix B provides an example illustrating the user

profile access interface.

5.3 Expressing Expert Advice Model Using XTM Syntax

As we have mentioned throughout this thesis, rather than defining a brand-new syntax to represent our expert advice model, we prefer to exploit existing standards -namely XTM- for this purpose. However, expressing our expert advice model using XTMs is not that straight-forward, as the model we propose does not exactly overlap with topic maps data structure, although it has been heavily inspired/influenced by it. In this section, we will begin with briefly identifying the common concepts in XTM and our model, and then focus on the points where our model essentially differs from the topic maps. For each such difference, we will propose an approach which makes use of the XTM syntax (defined in terms of [33]) as-is, and also allows to embed our own constructs. In this way, processing of XTM documents that encapsulates our expert advice model would be just the same as that of an ordinary XTM document, and would normally output a *topic-map graph* as specified in [31]. However, applications based on our model (such as the SQL-TC language we have defined before) should post-process the resulting graph (or internal structure, more generally) to further create some particular entities that are specific to our model. This approach is intended to provide that any XTM document that embodies constructs specific to our expert advice model would be safely used by any other XTM-based application. Similarly, our applications would be able to employ (almost) any existing XTM document as an expert advice repository, as long as a simple condition is satisfied, as it will be discussed below.

In general, topic maps and expert advice model share a common attitude in many cases, such as the notion of topics, metalinks, and source references (the latter two are named as *associations* and *occurrences* in topic map standard respectively, and used interchangeably throughout this section). To illustrate the correspondence, the (simplified) XTM syntax fragment for topic "Stephen King"

of type "novelist" in the domain of "literature" is given below. The fragment also implies that there is source of type "web-site" for this topic, located at "www.king.com". Topic id T1 is for internal processing and referencing purposes.

```
<topic id="T1">
        <instanceOf>
            <topicRef xlink:href="#novelist"/>
        </instanceOf>
        <baseName>
            <baseNameString>Stephen King</baseNameString>
            <scope>
                    <topicRef xlink:href="#literature"/>
            </scope>
        </baseName>
        <occurrence id="S1">
            <instanceOf>
                <topicRef xlink:href="#website"/>
            </instanceOf>
            <resourceRef xlink:href="http://www.king.com"/>
        </occurrence>
    </topic>
```

Now, let us assume another topic corresponding to novel "Carrie" which has the internal id T2. Then, the metalink instance "Stephen King"$\rightarrow$ *WrittenBy* "Carrie" is expressed as shown below.

```
   <association id="M1">
        <instanceOf>
            <topicRef xlink:href="#writtenBy"/>
        </instanceOf>
        <member>
            <roleSpec>
                <topicRef xlink:href="#author"/>
```

```
            </roleSpec>
            <topicRef xlink:href="#T1"/>
        </member>
        <member>
            <roleSpec>
                <topicRef xlink:href="#novel"/>
            </roleSpec>
            <topicRef xlink:href="#T2"/>
        </member>
    </association>
```

Thus, we roughly show that basic entities of our expert advice model such as topics (with a type and a domain), metalink instances and sources can be expressed by using the XTM elements <topic>, <instanceOf>, <association> and <occurrence>, along with their sub-elements or attributes (see Appendix D for the XTM syntax). However there are also some differences between the expert advice model and topic maps. We identify some of these differences, as well as the short-cut methods to handle them, as follows.

- In topic maps standard (and XTMs), associations are said to be *direction-less*, which states that the type of an association in a given association does not imply any directionality. For instance, in the above association instance "Stephen King"$\rightarrow$*WrittenBy* "Carrie", there are two player topics, and any processing application might (and should) deduce the sense of direction only by considering the player topics' role types, which are author and novel respectively. On the other hand, in this thesis, we mostly concentrate on *binary* metalink types (which still contribute to a very large set of metalinks). In this sense, to leverage/improve the system performance and ease the potential users' understanding, we suggest that a metalink instance should include two "system-specified" role types, namely "antecedent" and "consequent", which will provide a clue to the system about the semantics of this metalink instance.

  To provide the notion of directionality using the current XTM syntax, the

most straightforward method is to explicitly couple the player topics with "antecedent" or "consequent" types in every metalink instance. Thus, in the above example, topic "Stephen King" would be of role type "antecedent" and "Carrie" would be of role type "consequent". Alternatively, one can declare the role type information (in terms of types "antecedent" and "consequent") in the *metalink signature*, which removes the requirement of explicitly stating them in each metalink instance, and allows the use of other role types as in metalink instances. Consider the following XTM fragment declaring the signature of metalink type "*WrittenBy*". Note that, the signature itself is defined in terms of a metalink, where the metalink type is the system defined topic "*signature-template*". The Published Subject Indicators (PSIs) [32] for this metalink type and the system defined role types that exist in this metalink type could be published to support any other application that would like to use them in the same way we do.

```
<association id="signature#1">
        <instanceOf>
            <topicRef xlink:href="#signature-template"/>
        </instanceOf>
        <member>
            <roleSpec>
                <topicRef xlink:href="#metalink-type"/>
            </roleSpec>
            <topicRef xlink:href="#WrittenBy"/>
        </member>
        <member>
            <roleSpec>
                <topicRef xlink:href="#antecedent"/>
            </roleSpec>
            <topicRef xlink:href="#author"/>
        </member>
<member>
            <roleSpec>
                <topicRef xlink:href="#consequent"/>
```

```
            </roleSpec>
            <topicRef xlink:href="#novel"/>
        </member>
    </association>
```

For any conformant XTM processor, the above markup will denote an ordinary metalink instance, which might not be very informative or interesting. However, our post-processing application will interpret this markup as a metalink signature which is defined for the metalink type *WrittenBy* and enforcing that in all instances of *WrittenBy*, the topics that play the role of "author" should be interpreted as "antecedent", and the topics that play the role of "novel" should be interpreted as "consequent".

As the astute reader will realize, this implies the only requirement for our system to use an existing arbitrary XTM document. As long as metalink signatures are provided with the direction specifying role types in an XTM document, this particular XTM document may safely serve as an expert advice repository in our prototype system.

- In our expert advice model, we attach *importance values* to topic, metalink and source entities. To achieve this using XTM syntax, we declare three special metalink types: *HasTAdvice, HasMAdvice* and *HasSAdvice*. These metalink types have the signatures :

  HasTAdvice: importance-value → topic-entity
  HasMAdvice: importance-value → metalink-entity
  HasSAdvice: importance-value → source-entity

  The use of metalink instances conforming to the above metalink types are exemplified in the Appendix E.

- As another difference, the expert advice model of Chapter 3 allows topics to be attached a *maximum detail level*. Subsequently, topics that participate in a metalink instance might additionally specify detail levels as in "Diabetes Complications [2] → Prerequisite Diabetes[1]". Attaching maximum detail level to a topic is simply achieved by defining a metalink type with the signature *HasMaxDetail*: max-detail-value → topic-entity. Similarly, the

metalink type with signature *HasSDetail*: source-detail-value → source-entity is used to attach a detail level value to a source entity.

On the other hand, it is a little bit more verbose to make a topic with a particular detail level play in a metalink instance. For this purpose, we define yet another metalink type, called *MetalinkPlayerDetails*, whose players denote a metalink instance, and the attached detail-level values for "antecedent" and "consequent" players of this particular metalink instance. Note that this is possible in XTM, since an association is permitted to be a player in other associations [32]. Further note that, although the notation of the system-defined metalink type *MetalinkPlayerDetails* is ternary, the interpretation of it would still result in binary metalink instances. Thus, the above metalink instance example with detail levels is expressed as follows:

```
<association id="preq1">
        <instanceOf>
            <topicRef xlink:href="#prerequisite"/>
        </instanceOf>
        <member>
            <roleSpec>
                <topicRef xlink:href="#antecedent"/>
            </roleSpec>
            <topicRef xlink:href="#diabetes"/>
        </member>
        <member>
            <roleSpec>
                <topicRef xlink:href="#consequent"/>
            </roleSpec>
            <topicRef xlink:href="#diabetesComplications"/>
        </member>
    </association>

<association id="preq-util-1">
        <instanceOf>
```

```
            <topicRef xlink:href="#metalinkPlayerDetails"/>
        </instanceOf>
        <member>
            <roleSpec>
                <topicRef xlink:href="#metalink-instance"/>
            </roleSpec>
            <topicRef xlink:href="#preq1"/>
        </member>
        <member>
            <roleSpec>
                <topicRef xlink:href="#antecedent"/>
            </roleSpec>
            <topicRef xlink:href="#2"/>
        </member>
        <member>
            <roleSpec>
                <topicRef xlink:href="#consequent"/>
            </roleSpec>
            <topicRef xlink:href="#1"/>
        </member>
    </association>
```

Please note that we also realize that our system-defined metalink types make the notation more verbose. Instead, the above extensions can be provided more simply by adding a few more elements or attributes to the XTM DTD, which would result in defining a non-conformant syntax. However, as we plan to make our system flexible enough to use any existing XTM document, we preferred to use the above constructs whenever required. Note that the use of these system-defined metalink types (except for signatures, as discussed before) is *not* mandatory, i.e., an XTM specifying no detail levels for its topics would not need to use the *MetalinkPlayerDetails* metalink type. Thus, we may safely use existing XTM documents, as long as they provide the notion of direction either explicitly or through the use of metalink signatures. Similarly, the XTMs enriched with our constructs

may be used by any other XTM-processing application as well. The XTM document provided in Appendix E makes use of these constructs whenever convenient.

5.4 XTM Processor Implementation

We have implemented a general-purpose XTM Processor which is conformant to [31] and may process any valid XTM document. The general class hierarchy is inspired from a freely available topic map processor [17], but extends it in a number of ways according to the rules given in the processing model specification [31]. Basically, the input to our processor is a valid XTM document and the output is an in-memory representation of the topic map graph, which is stored in terms of Java HashTable data objects at the moment.

In essence, the processor functions as follows: At first, an input XTM document is parsed using the DOM API [29] for XML parsing and an internal topic map graph is generated, by creating nodes and arcs required by XTM elements, either explicitly or implicitly. The graph includes *t-nodes, a-nodes* and *s-nodes,* for *topic, association* and *scope* elements respectively. In accordance with [31], a-nodes are defined as a subclass of t-nodes, which allows an association instance to be a player in another association instance. The topic map graph is stored in main memory (at the moment) similar to *adjacency-list* representation of graphs. Each node keeps track of what other nodes it is connected to, and with what type of arcs. The nodes are represented as objects in the system with unique internal ids. The type information for an arc is captured by the name of the field in the object. As an example, an a-node representing an association instance is connected to t-nodes (for its player topics) via "association member" arcs. In this case, the array of *association players* could include a list of *t-ids.* However, an arc would be further labeled with the role that a particular topic plays in the association (e.g., author, novel), so instead of storing only t-node ids we store instances of class AssociationPlayer which has a t-node id and the role it acts.

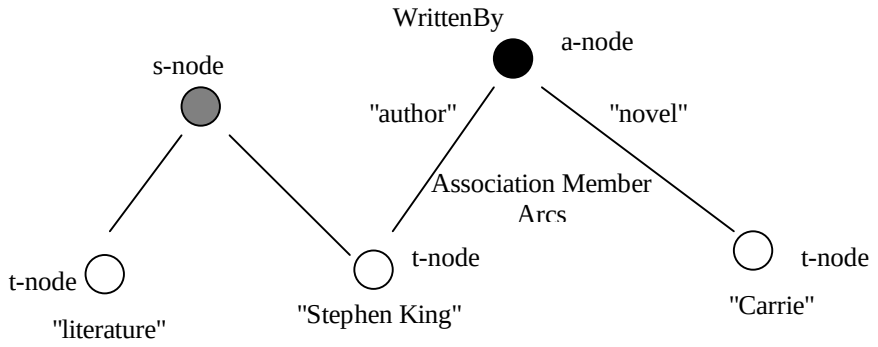

Figure 5.2: A simple graph representing the association (metalink instance) "Stephen King" $\rightarrow$ *WrittenBy* "Carrie" along with role types and domain information

A (very) simplified graph fragment is given in Figure 5.2. for the purpose of illustration. (Note that, the processing model indeed enforces that properties of a topic, just like its type, name and occurrences, should also be expressed in terms of associations in the graph, which makes it look much more complex).

All the t-nodes and a-nodes are kept in a *HashTable*, just like the s-nodes that are kept in a separate table. Additionally, we create hash tables indexed with *subject indicator resources*, to deduce and merge the topics which represent the same real life subject. Thus, the processor implements *subject-based topic merging* mechanism. Name-based topic merging mechanism, which is easier, is not implemented due to time limitations.

Finally, please note that, the XTM processing model [31] is still an effort under development, which implies that some of the features discussed here may be obsolete in the future. Besides, a number of issues about the topic map graphs discussed above were not resolved at the time of writing this thesis, which made us to determine our own design choices. Again, these are subject to change in accordance with the stabilization of the processing model. In any case, we envision that the modular architecture of our system would help, as any modification required in the implementation of the processor would be localized, and will not be propagated to other components, most probably.

class Topic [String *topic-id*, String *name*, String *type*, String[] *domain*, String *author*, Int *max-detail*, TSRef [] *sources*, int *TAdviceVal*]

class TSRef [String *source-id*, String *address*, int *detail-level*, String *role*, String *mediatype*, String *LastUpdated*, int *SAdvicVal*]

class Metalink [String *metalink-id*, String *m-type*, Topic *antecedent-player*, int *ante-detail-level*, Topic *consequent-player*, int *cons-detail-level*, String *m-domain*, int *MadviceVal*]

class MetalinkSignature [String *signature-id*, String *m-type*, String *antecedent-role-type*, String *consequent-role-type*]

Figure 5.3: Expert Advice Model Classes

5.5 Post-processing Internal XTM Representation

As it is mentioned before, we have a *post-processing stage* over the internal XTM graph to create instances of the expert advice model as defined in Chapter 3. Although this stage may seem rather costly, one should note that the topic map graph itself (to the extent it is specified in [31]) is a quite cumbersome data structure to process, and to query. (For instance, even to reach the name of a topic node, one should follow the *baseName association* link). Thus, the post-processing stage serves in two major ways: (i) allows us to create a rather compact and simpler data structure in terms of expert advice model entities, and (ii) allows us to extract and process *application specific* information that is embedded via the constructs of Chapter 5.3.

In accordance with Chapter 3.2, we define the expert advice object model in terms of the classes in Figure 5.3. Note that, in the prototype implementation, we assume that for each topic entity exactly one name and type is provided in a given (set of) domain, and furthermore all metalink types are binary, where both players are single topics.

Thus, once the topic map graph is created, the post-processing stage will traverse it and create the instances of our data model. In Appendix A, the

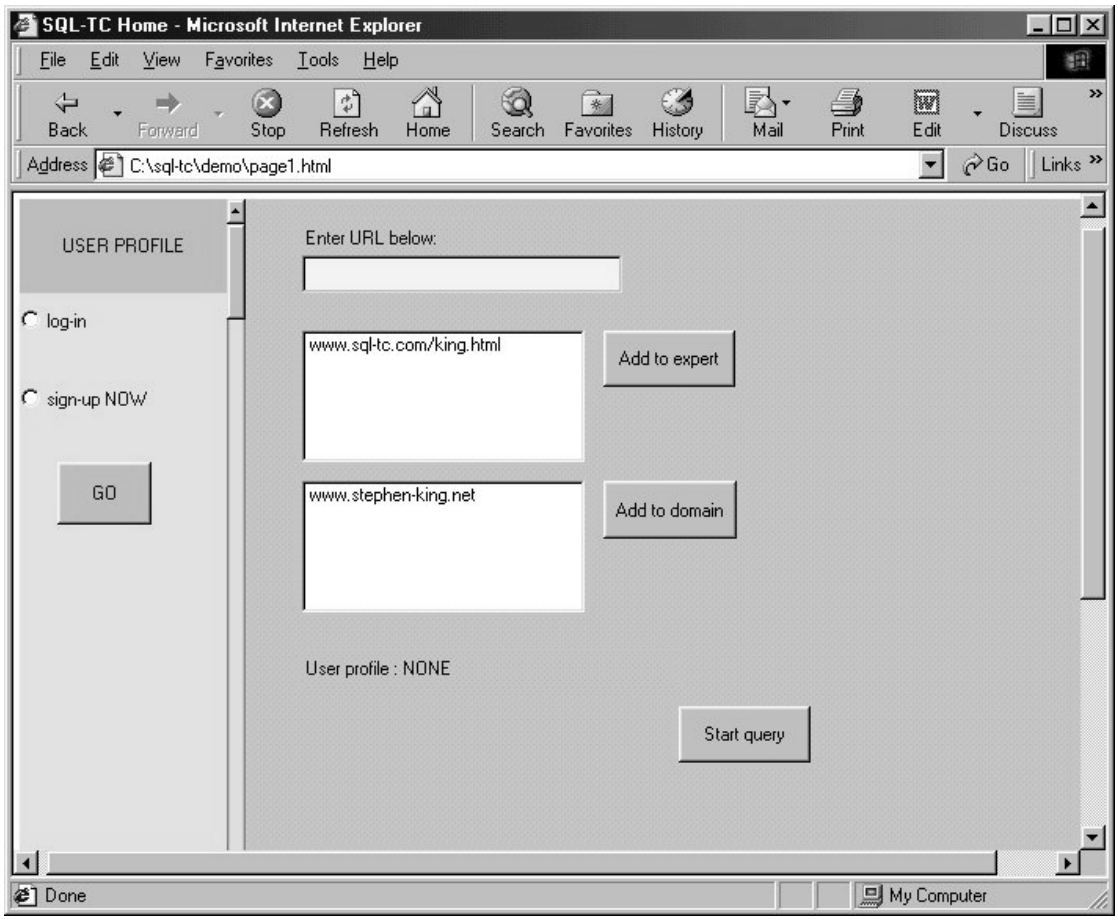

Figure 5.4: Visual Query Interface

instances of these classes are shown in a tabular manner, in accordance with the prototype implementation which stores the objects into a relational database system, namely MS Access. Note that, an object-relational database management system should better be used to support *set-valued* attributes, which are flattened in the prototype implementation.

5.6 SQL-TC Queries and Visual Interface

Visual query interface is implemented as a Java servlet, which is available at [15]. When a user is first connected to our site, (s)he has the option to login in order to access local user profile (Figure 5.4). Then, the user specifies the

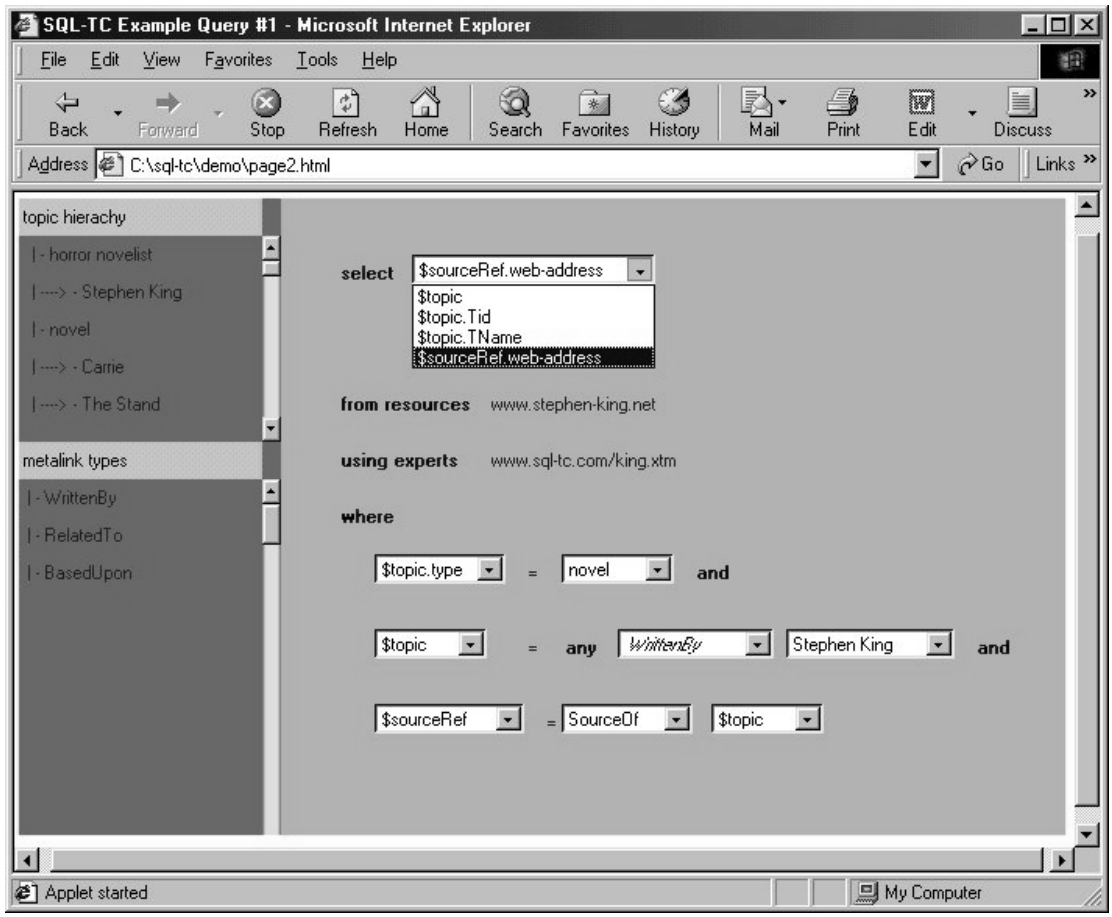

Figure 5.5: Query Design Interface

address of the web domain(s) to be queried and/or one or more XTM expert advice repositories. In the restricted demo in [15], we provide a (virtual) domain and an XTM document of Appendix E as defaults.

Next, query design page (Figure 5.5) is presented where the user may pose queries by using the topic and metalink objects available in the specified advice repositories. At the moment, we provide a web page with a *query template* that allows designing queries of one particular type. Providing a functional and user-friendly GUI supporting all features of SQL-TC will need more work and will be reported later.

In the prototype demo [15], we return query outputs as interactive tables, where the metadata objects and/or the sources in the tables are hyperlinks that

may be followed and navigated.

In the current system, an SQL-TC query is translated into an equivalent SQL query and evaluated over a relational database system. In Appendix A and B, entities of example domain are represented in a tabular manner similar to the actual relational representation where all set values (e.g., TDomain) are flattened. At the moment, ranking of the output tuples is handled at the application level (i.e., by the use of cursors) whereas we plan to adapt the strategy of [53] in our query processing engine. In the engine, we also plan to support and evaluate similarity-based queries (such as in Example 4.2.4 of Chapter 4) by algorithms specifically tailored for them. The work for the query processing algorithms is currently proceeding [76].

Chapter 6

Conclusion and Future Work

In this thesis, we develop a web information space model and its query language to allow sophisticated queries/searches over web resources. The proposed information model has three major components: (i) information resources that are representing web-based documents, (ii) expert advice model that is contributing to a *topic-centric knowledge index* over the resources, and (iii) personalized information model that is capturing user preferences and knowledge. In this study, information resources are assumed to be HTML/XML documents and the user profile is captured in XML documents. XTMs are used to serve as expert advice repositories that are imposed over information resources. Expert advice repositories contribute to a *semantic index* as they identify the *topics* and their relationships (*metalinks*) in a resource set, and provide links to the actual *occurrences* of topics in these resources. 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.

The SQL-like query language SQL-TC is designed to operate over this information space and query the information resources by incorporating the expert advices and users' personalized information. SQL-TC queries are expressed using topics, metalinks and their sources, and they are capable of querying both the associated information resources and the expert advice repositories themselves in an integrated manner. In this sense, SQL-TC provides multi-database access

as well as heterogeneous data access. One distinguishing feature of SQL-TC is the concept of *topic closures*, which retrieves the set of all topics that are logically implied by the queried topic(s) with respect to a particular metalink type. Pruning topic closure sets by consulting user knowledge -that is kept in the user profile- is possible, which may otherwise grow too large. Furthermore, query outputs may be ranked with respect to importance values supplied in the expert advice model, allowing SQL-TC to return the most relevant tuples/objects as a response. As SQL-TC is capable of using multiple expert advice repositories simultaneously along with personalized information databases, the issue of expert advice *conflicts* and their resolution is important. In this thesis, we discuss a user-preference based resolution approach, which may further be improved in the future.

Finally, the thesis presents the system architecture and a prototype implementation using XTMs as expert advice repositories. In this sense, our work serves as a real-life application of XTM effort and may also contribute to identify and resolve many issues with the currently evolving XTM standard.

Future work will include the following issues:

- Developing semi-automatic tools to gather metadata information into expert advice repositories for very large web domains (as discussed in Chapter 3.4),
- Developing a sophisticated GUI to allow the use of SQL-TC by naive web-surfers,
- Developing a large-scale fully-functional system implementation to allow ordinary users to evaluate the system performance in terms of response time and quality -which can be measured by recall and precision metrics,
- Providing an interactive refinement mechanism for user preferences which is based on the user's feedback about the relevancy of returned results,
- Developing query processing and optimization algorithms for SQL-TC queries, especially tailored for *approximate (similarity-based) queries* and

topic closure computations. The work for this purpose is actually underway and will be reported soon.

Bibliography

[1] Hypermedia/time-based structuring language (HyTime) users' group home page, www.hytime.org.

[2] IETF draft standard for uniform resource identifiers (URIs), http://www.w3.org/addressing/#9808uri.

[3] Infoloom home page, http://www.infoloom.com.

[4] Java topic map engine, http://thinkalong.com/jtme/jtme.html.

[5] The K42 topic map engine, http://k42.empolis.co.uk.

[6] Microsoft moneycentral, http://moneycentral.msn.com/home.asp.

[7] Microsoft MSDN online support, http://support.microsoft.com/servicedesks/msdn.

[8] Mondeca home page, http://www.mondeca.com.

[9] Online collections of the Smithsonian Institution, http://www.si.edu.

[10] Ontopia topic map techonology home page, http://www.ontopia.net/atlas.

[11] Open eBook forum web site, http://www.openebook.com.

[12] SAX 2.0: The simple API for XML, http://www.megginson.com/sax/index.html.

[13] SigmaLink: The intranet based document management, http://www.step.de/sigmalink.htm.

[14] Simple object access protocol (SOAP) 1.1, `http://www.w3.org/tr/soap/`.

[15] SQL-TC servlet, `http://139.179.21.49:8080/servlet/sql-tc`.

[16] tmproc: A topic map engine, `http://www.infotek.no/~grove/software/tmproc/index.html`.

[17] Topic maps for java (TM4J) engine, `http://www.techquila.com`.

[18] Topicmaps.net web site, `http://www.topicmaps.net`.

[19] Unified modeling language (UML), version 1.3, `http://www.omg.org/technology/documentsformal/uml.htm`.

[20] XML cover pages: Topic maps, `http://www.oasis-open.org/cover/topicmaps.html`.

[21] XML linking language (XLink) version 1.0, `http://www.w3.org/tr/xlink/`.

[22] XML pointer language (XPointer) version 1.0, `http://www.w3.org/tr/xptr`.

[23] XSLT stylesheets for converting ISO 13250 topic map documents into XTM 1.0 syntax, `http://www.cogx.com/xslt4tm2xtm.html`.

[24] XTM mailing list, `www.yahoogroups.com/list/xtm-wg`.

[25] (XTM) topicmaps.org web site, `http://www.topicmaps.org`.

[26] XTM working documents, `http://www.doctypes.org/xtm/home.html`.

[27] XML-data: World wide web consortium (W3C) note, january 1998. Available at `http://www.w3.org/TR/1998/NOTE-XML-data-0105/`.

[28] ACM SIGMOD workshop on the web and databases (WebDB'99), available at `http://www.acm.org/sigmod/dblp/db/conf/webdb/webdb1999.html#ludascherpv99`, june 1999.

[29] The WWW consortium (W3C's) DOM (document object model) web page, `http://www.w3.org/dom/`, 1999.

[30] ECDL workshop on the semantic web, september 2000. Electronic proceedings available at `http://www.ics.forth.gr/proj/isst/SemWeb`.

[31] XTM processing model 1.0, topicmaps.org AG review specification, `http://topicmaps.org/xtm/1.0/xtmp1.html`, december 2000.

[32] XML topic maps (XTM) 1.0 specification, `http://topicmaps.org/xtm/1.0`, 2001.

[33] XTM DTD v1.0, `http://topicmaps.org/xtm/1.0`, 2001.

[34] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann Publishers, San Francisco, 2000.

[35] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J. Wiener. The Lorel query language for semistructured data. *Int. Jour. on Digital Libraries*, 1(1):68–88, 1997.

[36] R. Agrawal and E. Wimmers. A framework for expressing and combining preferences. In *ACM SIGMOD Conf.*, pages 297–306, 2000.

[37] K. Ahmed. Topic maps for repositories. In *Proc. of XML Europe 2000*, Paris, June 2000. Available at `http://www.infoloom.com/gcaconfs/WEB/paris2000/S29-04.HTM#N10`.

[38] L. Alschuler. News report on extreme markup languages conf., 2000. The news archieve available at `http://www.xml.com`.

[39] I. S. Altıngövde, S. Özel, Ö. Ulusoy, G. Özsoyoğlu, and Z. M. Özsoyoğlu. Topic-centric querying of web-based information resources. In *Proc. of the 12th International Conference on Database and Expert Systems Applications (DEXA) 2001*, pages 699–711, Munich, Germany, 2001.

[40] G. Arocena and A. Mendelzon. WebOQL: Restructuring documents, databases and webs. In *Proc. of the Int. Conf. on Data Engineering (ICDE)*, pages 24–33, Orlando, Florida, 1998.

[41] C. Baird. Topic map cartography: A discussion of topic map authoring. In *Proc. of XML Europe 2000*, Paris, June 2000. Available at `http://www.infoloom.com/gcaconfs/WEB/paris2000/S11-03.HTM#N1`.

[42] Z. Bar-Yossef, Y. Kanza, Y. A. Kogan, W. Nutt, and Y. Sagiv. Querying semantically tagged documents on the world-wide web. In *Next Generation Information Technologies and Systems*, pages 2–19, 1999.

[43] T. Berners-Lee. Semantic web roadmap, W3C draft, available at `http://www.w3.org/designissues/semantic.html`, january 2000.

[44] M. Biezunski, M. Bryan, and S. Newcomb. ISO/IEC 13250 topic navigation maps, `http://www.ornl.gov/sgml/sc34/document/0058.htm`, 1999.

[45] M. Biezunski and C. Hamon. A topic map of this conference's proceedings. In *Proc. of Third GCA International HyTime Conference*, Seattle, Washington, USA, August 1996. Available at `http://www.infoloom.com/IHC96/mb214.htm`.

[46] A. Bonifati and S. Ceri. Comparative analysis of five XML query languages. *SIGMOD Record*, 29(1):68–79, 2000.

[47] A. Bosworth and A. L. B. Jr. Microsoft's vision for XML. *IEEE Data Engineering Bulletin*, 22(3):35–43, 1999.

[48] T. Bray, J. Paoli, and C. Sperberg-McQueen. Extensible markup language 1.0 specification, world wide web consortium (W3C), February 1998.

[49] D. Brickley and e. R.V. Guha. Resource description framework (RDF) schema specification, W3C proposed recommendation 03 march 1999. The latest version is available at `http://w3.org/TR/PR-rdf-schema`.

[50] J. Broekstra, M. Klein, S. Decker, D. Fensel, and I. Horrocks. Adding formal semantics to the web building on top of RDF schema. In *Proc. of ECDL Workshop on the Semantic Web*, 2000. Available at `http://www.ics.forth.gr/proj/isst/SemWeb/proceedings/session2-2`.

[51] P. Buneman. Semistructured data. In *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 117–121, Tucson, Arizona, May 1997.

[52] P. Buneman, S. Davidson, G. Hillebrand, and D. Suciu. A query language and optimization techniques for unstructured data. In *Proceedings of ACM-SIGMOD International Conference on Management of Data*, pages 505–516, Montreal, Canada, june 1996.

[53] M. J. Carey and D. Kossmann. On saying "enough already!" in SQL. In *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data*, pages 219–230, Tucson, Arizona, 1997.

[54] V. Christophides. Community webs (C-Webs): Technological assessment and system architecture. Technical Report D5, September 2000.

[55] I. Cingil, A. Dogac, and A. Azgın. A broader approach to personalization. *ACM Comm.*, 43(8):136–141, august 2000.

[56] L. Delcambre, D. Maier, R. Reddy, and L. Anderson. Structured maps: Modeling explicit semantics over a universe of information. *L., Journal of Digital Libraries*, 1(1), 1997.

[57] A. Deutsch, M. F. Fernandez, D. Florescu, A. Y. Levy, and D. Suciu. A query language for XML. *WWW8 / Computer Networks*, 31(11-16):1155–1169, 1999.

[58] A. Deutsch, M. F. Fernandez, and D. Suciu. Storing semistructured data with STORED. In *SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data*, pages 431–442, Philadephia, Pennsylvania, june 1999.

[59] M. F. Fernandez, D. Florescu, A. Y. Levy, and D. Suciu. A query language for a web-site management system. *SIGMOD Record*, 26(3):4–11, 1997.

[60] D. Florescu, A. Y. Levy, and A. O. Mendelzon. Database techniques for the world-wide web: A survey. *SIGMOD Record*, 27(3):59–74, 1998.

[61] H. Folch. Semantic tagging of a corpus using the TNM standard. Technical report, Electricit de France (Division Recherche et Dvelopement).

[62] E. Freese. Topic maps vs. RDF. In *Proc. of (GCA) Extreme Markup Languages 2000*, 2000. Available at `http://ep.open.ac.uk/PubSys/resources/html/free0000.html`.

[63] E. Freese. Using topic maps for the representation, management & discovery of knowledge. In *Proc. of XML Europe 2000*, Paris, June 2000. Available at `http://www.infoloom.com/gcaconfs/WEB/paris2000/S22-01.HTM#N2`.

[64] G. Gronmo. Creating semantically valid topic maps. In *Proc. of XML Europe 2000*, Paris, June 2000. Available at `http://www.infoloom.com/gcaconfs/WEB/paris2000/S29-02.HTM#N1`.

[65] J.-R. Gruser, L. Raschid, M. E. Vidal, and L. Bright. Wrapper generation for web accessible data sources. In *Proceedings of the 3rd IFCIS International Conference on Cooperative Information Systems (CoopIS)*, pages 14–23, New York, USA, 1998.

[66] J. Han, O. Zaiane, and Y. Fu. Resource and knowledge discovery in global information systems: A scalable multiple layered database approach. In *Proc. of Conf. on Advances in Digital Libraries*, Washinton, DC, May 1995.

[67] E. Kimbler. A tutorial introduction to SGML architectures. Available at `http://www.isogen.com/papers/archintro.html`.

[68] D. Konopnicki and O. Shmueli. W3QS: A query system for the world wide web. In *Proc. of the Int. Conf. on Very Large Data Bases (VLDB)*, pages 54–65, Zurich, Switzerland, 1995.

[69] R. Ksiezyk. Answer is just a question [of matching topic maps]. In *Proc. of XML Europe 2000*, Paris, June 2000. Available at `http://www.infoloom.com/gcaconfs/WEB/paris2000/S22-03.HTM#N2`.

[70] O. Lassila and R. R. Swick. Resource description framework (RDF) model and syntax specification, W3C recommendation 03 february 1999. The latest version is available at `http://w3.org/TR/REC-rdf-syntax`.

[71] A. Y. Levy, A. Rajaraman, and J. J. Ordille. Querying heterogeneous information sources using source descriptions. In *VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases*, pages 251–262, Bombay, India, 1996.

[72] D. Maier and L. Delcambre. Superimposed information for the internet. In *(Informal) Proc. of WebDB*, pages 1–9, 1999.

[73] A. Mendelzon, G. Mihaila, and T. Milo. Querying the WWW. *International Journal on Digital Libraries*, 1(1):54–67, april 1997.

[74] G. Moore. RDF and topic maps an exercise in convergence. In *Proc. of XML Europe 2001*, Berlin, 2001. Available at `http://www.topicmaps.com/topicmapsrdf.pdf`.

[75] S. Newcomb and M. Biezunski. Topic maps go XML. In *Proc. of XML Europe 2000*, Paris, June 2000. Available at `http://www.infoloom.com/gcaconfs/WEB/paris2000/S11-02.HTM#N1`.

[76] G. Özsoyoğlu. Sideway value propogating algebra (SVA). Working draft, 2001.

[77] G. Özsoyoğlu, M. Anderson, and Z. M. Özsoyoğlu. Web search with metadata links and multimedia presentations. In *In Proc. of the Sixth Int. Workshop on Multimedia Information Systems*, Chicago, USA, October 2000. Available at `http://nashua.cwru.edu/TOpapers/MIS2000.pdf`.

[78] G. Özsoyoğlu, N. Balkir, G. Cormode, and Z. M. Özsoyoğlu. Electronic books in digital libraries. In *Proc. of the IEEE Advances in Digital Libraries (ADL) Conf.*, pages 5–14, May 2000.

[79] S. Pepper. Topic maps and RDF: A first cut. Available at `http://www.ontopia.net/topicmaps/learn_more.html`.

[80] S. Pepper. Euler, topic maps, and revolution. In *Proc. of XML Europe 99*, 1999. Available at `http://www.infoloom.com/gcaconfs/WEB/granada99/pep.HTM`.

[81] S. Pepper. The TAO of topic maps: finding the way in the age of infoglut. In *Proc. of XML Europe 2000*, Paris, June 2000. Available at `http://www.infoloom.com/gcaconfs/WEB/paris2000/S11-01.HTM#N2`.

[82] H. Rath. Topic maps self control. In *Proc. of (GCA) Extreme Markup Languages 2000*, 2000. Available at `http://ep.open.ac.uk/PubSys/resources/html/rath0315.html`.

[83] H. Rath and S. Pepper. *Topic Maps at Work. Chapter 1 in: XML Handbook.* Prentice Hall, 2. edition, 1999.

[84] J. Robie. The design of XQL, `http://www.texcel.no/whitepapers/xql-design.html`, 1999.

[85] C. Schmidt. A proposal for a TMQL based on OQL and an underlying object model of XTM. Working Draft, 2001.

[86] J. Ullman. *Principles of Database and Knowledge-Base Systems*, volume 1. Computer Science Press, 1998.

[87] O. Zaiane and J. Han. Resource and knowledge discovery in global information systems: A preliminary design and experiment. In *Proc. of Conf. on Knowledge Discovery and Data Mining*, pages 331–336, Montreal, Canada, 1995.

[88] O. R. Zaiane and J. Han. WebML: Querying the world-wide web for resources and knowledge. In *ACM CIKM'98 Workshop on Web Information and Data Management (WIDM'98)*, pages 9–12, Washington DC, 1998.

Appendix A

Expert Advice Repositories

In the following, we provide two expert advice repositories that are assumed to be available at the virtual web locations www.sql-tc.com/king.xtm and www.horror-books.com/books.xtm. The XTM files are processed as described in Chapter 5.2 to create the below tabular forms. (A fragment of the original XTM file is provided in Appendix E). Each row in the tables corresponds to an instance of one of the expert advice model classes as defined in Chapter 5. The former web-site provides expert advices about the horror writer Stephen King and includes a number of concepts and relationships about Stephen King himself, his books, movies based on his work, etc. and provides pointers to various kinds of resources for these concepts (topics) or relationships (metalinks)[1]. The latter web site is a more general one including expert advices about horror writers, books, characters of books, etc.

[1] Please note that, most of the novels, characters, web locations, etc. given in this Appendix and in the examples throughout this thesis are imaginary.

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

Tid	TDet. level	TType	Tname	TDomain	TAdv.	Source
T1	-	horror novelist	"Stephen King"	literature	1	{S1, S2}
T2	-	novel	"Carrie"	literature	1	{S3, S4}
T3	-	novel	"The Stand"	literature	0.8	{S5, S6}
T4	-	movie	"Story of Carrie"	cinema	0.7	-
T5	-	musical	"Carrie: The Musical"	leisure	0.2	-
T6	4	novel	"Wizard and The Glass"	literature	0.3	-
T7	3	novel	"The Wasteful Lands"	literature	0.4	S7
T8	2	novel	"Drawings of Three"	literature	0.6	S8
T9	1	novel	"Dark Tower"	literature	0.8	-
T10	-	movie director	"John Carpenter"	cinema	-	-

Table A.1: Topics of Expert E1

Sign-id	Mtype	Antecedent-role	Consequent-role
SG1	WrittenBy	Author	Novel
SG2	BasedUpon	Base-topic	Result-topic
SG3	RelatedTo	Topic	Topic
SG4	InfluencedBy	Influences	Influenced
SG5	NovelCharacters	Novel	Character
SG6	NovelsOfNovelCharacters	Character	Novel

Table A.2: Metalink Signatures of Expert E1

In Table A.3, player topics in a metalink instance are identified with the triple TName, TType, TDomain to enhance the readability. In the actual class definition (and prototype implementation) of Chapter 5, the player topics will be referenced by using their internal ids. For instance, the antecedent player is T1 and the consequent player is T2 for the first metalink instance M1 of Table A.3.

Mid	MType	MDomain	Antecedent players	ADL	Consequent players	CDL	MAdv
M1	WrittenBy	{literature, horror}	[Stephen King, horror novelist, literature, -]	-	[Carrie, novel, literature,-]	-	1
M2	WrittenBy	{literature, horror}	[Stephen King, horror novelist, literature, -]	-	[The Stand, novel, literature, -]	-	0.6
M3	WrittenBy	{literature, horror}	[Stephen King, horror novelist, literature, -]	-	[Wizard and The Glass, novel, literature, -]	-	0.6
M4	WrittenBy	{literature, horror}	[Stephen King, horror novelist, literature, -]	-	[The Wasteful Lands, novel, literature, -]	-	0.6
M5	WrittenBy	{literature, horror}	[Stephen King, horror novelist, literature, -]	-	[Drawings of Three, novel, literature, -]	-	0.6
M6	WrittenBy	{literature, horror}	[Stephen King, horror novelist, literature, -]	-	[Dark Tower, novel, literature, -]	-	0.6
M7	BasedUpon	-	[Carrie, novel, literature, -]	-	[Story of Carrie, movie, cinema, -]	-	0.6
M8	BasedUpon	-	[Story of Carrie, movie, cinema, -]	-	[Carrie: The Musical, musical, -]	-	-
M9	RelatedTo	-	The Wasteful Lands, novel, literature, 3]	3	[Wizard and The Glass, novel, literature, 4]	4	0.3
M10	RelatedTo	-	[Drawings of Three novel, literature, 2]	2	[The Wasteful Lands, novel, literature, 3]	3	0.5
M11	RelatedTo	-	[Dark Tower, novel, literature, 1]	1	[Drawings of Three novel, literature,2]	2	0.6
M12	Novel-Characters	-	[The Stand, novel, literature, 1]	1	[Jack Park, hero, novel characters, -]	-	0.7
M13	InfluencedBy	-	[Jack Park, hero, novel characters, -]	-	[John Smith, character, novel characters, -]	-	0.8
M14	RelatedTo	-	[Story of Carrie, movie, cinema, -]	-	[John Carpenter, movie director, cinema, -]	-	-
M15	RelatedTo	-	[Carrie: the Musical, musical, leisure, -]	-	[Broadway Season 1980, musical, leisure, -]	-	-

Table A.3: Metalinks of Expert E1

Sid	Web-address	Role	Media Type	Last Updated	Detail level	SAdv.
S1	www.king.com/	Website	multimedia	16.01.2001	-	1
S2	www.newsweek.com/ 17-10-2000/tragic-accident.html	Mentions	Text	20.01.2001	-	0.7
S3	www.books.com/ carrie.html	Summary	Text	-	-	0.5
S4	www.critics.com/ carrie.html	Review	Text	-	4	0.8
S5	www.books.com/ stand.html	Summary	Text	-	-	0.4
S6	www.critics.com/ stand.html	Review	Text	-	4	0.7
S7	www.critics.com/ dark3.html	Review	text	-	4	0.8
S8	www.critics.com/ dark2.html	Review	Text	-	4	0.3

Table A.4: Sources of Expert E1

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

Tid	TDetail level	TType	Tname	TDomain	T-Advice	Source
T11	-	novel	"Scream"	literature	0.3	S9
T12	-	novel	"Maniac"	literature	0.4	S10
T13	-	hero	"Jack Park"	novel characters	-	-
T14	-	character	"John Smith"	novel characters	-	-

Table A.5: Topics of Expert E2

Sign-id	Mtype	Antecedent-role	Consequent-role
SG7	InfluencedBy	Influences	Influenced
SG8	NovelsOfNovelCharacters	Character	Novel

Table A.6: Metalink Signatures of Expert E2

Mid	MType	MDom	Antecedent players	ADL	Consequent players	CDL	MAdv
M16	InfluencedBy	-	[Jack Park, hero, novel characters, -]	-	[John Smith, character, novel characters, -]	-	No
M17	NovelsOfNovel Characters	-	[John Smith, character, novel characters, -]	-	[Scream, novel, literature, -]	-	0.6
M18	NovelsOfNovel Characters	-	[John Smith, character, novel characters, -]	-	[Maniac, novel, literature, 1]	-	0.2

Table A.7: Metalinks of Expert E2

Sid	Web-address	Role	MediaType	LastUpdated	Detail level	S-Advice
S9	www.books.com/ scream.html	Summary	text	12.02.2001	-	0.6
S10	www.books.com/ maniac.html	Summary	text	13.02.2001	-	0.7

Table A.8: Sources of Expert E2

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 the 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

TName	**Detail level**	**Source address**	**Source role**	**Source media type**	**First visit**	**Last visit**	**Visit No**
"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	text	-	-	3

Table B.1: User-Knowledge (U)

Appendix C

BNF for SQL-TC

<SQL-TC_query> := <select_clause>
<from_clause>
<where_clause>
[<order_by_clause>]
| <SQL-TC_query> **union** <SQL-TC_query>
| <SQL-TC_query> **minus** <SQL-TC_query>
| <SQL-TC_query> **intersect** <SQL-TC_query>

<select_clause> := **select** [<topic_metalink_source_variable>] [**as** <variable>]
| **select** <aggregate_function>

<topic_metalink_source_variable> := <a_variable>[.<attribute>]
| <a_variable>[.<attribute>], <topic_metalink_source_variable>

<a_variable> := $<identifier>

<attribute> := <identifier>

<variable> := <letter>

<underscore> := -_

<identifier> := <letter>{<letter> | <digit> | <underscore>}

<letter> := **A..Z** | **a..z**

<digit> := **0..9**

<aggregate_function> := <aggregate_function_name>(<argument>)

<aggregate_function_name> := **max** | **min** | **avg** | **sum** | **count**

<argument> := <a_variable>[.<attribute>] | <importance_functions>

<importance_functions> :=
TAdviceMatch(<expert_var>,(<a_variable>[.<attribute>] | <string_constant>)) |
MAdviceMatch(<expert_var>,(<a_variable>[.<attribute>] | <string_constant>)) |

SAdviceMatch(<expert_var>,(<a_variable>[.<attribute>]|<url>))

<built_in_function> :=

<importance_functions> |

GetSourceAddresses(<profile_var>.**UserKnowledge**) |

GetLastVisitedDays(<profile_var>.**UserKnowledge**, <a_variable>.**web-address**) |

Contains(<a_variable>,<string_constant>) |

MetalinksWithTopic(<topic_name>,<topic_type>,<topic_domain>, <expert_name>)

<from_clause> := [**from resources** <url_list>]
[**using experts** <expert_list>]
[**with user profile** <profile_list>]

<url_list> := <url> | <url>,<url_list>

<url> := a valid URL

<expert_list> :=<url> [**as** <expert_var>]
| <url> [**as** <expert_var>],<expert_list>

<expert_var> := **E**{<digit>}

<profile_list> := <url> [**as** <profile_var>]
| <url> [**as** <profile_var>],<profile_list>

<profile_var> := **U**{<digit>}

<where_clause> := **where** <variable_declaration_clause> **and** <condition_clause>

<variable_declaration_clause> :=
<topic_variable_declaration>[**and** <variable_declaration_clause>]
| <metalink_variable_declaration>[**and** <variable_declaration_clause>]

<topic_variable_declaration> :=
<topic_var>{,<topic_var>} **in** <expert_var>.**Topics**
| <topic_var>{, <topic_var>} **in** (<expert_var>{,<expert_var>}).**Topics**

<metalink_variable_declaration> := (<metalink_var>|<metalink_type>)
{, (<metalink_var>|<metalink_type>)} **in** <expert_var>.**Metalinks**
|(<metalink_var>|<metalink_type>)
{, (<metalink_var>|<metalink_type>)} **in** (<expert_var>{,<expert_var>}).**Metalinks**

<topic_var> := <a_variable>

<metalink_var> := <a_variable>

<metalink_type> := <identifier> | <identifier>*

<condition_clause> := <metalink_closure_clause>
| <source_of_clause>
| <a_condition_clause>
| <SQL-TC_query>
| <condition_clause> **and** <condition_clause>
| <condition_clause> **or** <condition_clause>

<metalink_closure_clause> := <topic_var> =**any** (<metalink_closure_expression>)

<metalink_closure_expression> :=

<metalink_closure_expression> **or** <metalink_closure_term>
| <metalink_closure_term>
<metalink_closure_term> := <metalink_closure_term> **and** <metalink_closure_factor>
| <metalink_closure_factor>
<metalink_closure_factor> := (<metalink_closure_expression>)
| <metalink_closure_element>
<metalink_closure_element> := [<bool_not>] <metalink_type>(<closure_arguments>)
| ([<bool_not>] <metalink_type>(<multi_metalink_closure>))
<multi_metalink_closure> := [<bool_not>] <metalink_type>(<multi_metalink_closure>)
| [<bool_not>] <metalink_type>
<closure_arguments> :=
<topic_name>, <topic_type>, <topic_domain>, <expert_name>, <detail_level>
| ([<bool_not>] <metalink_type> | **SimTName**) (<closure_arguments>)
<topic_name> := "<identifier>" | ϵ
<topic_type> := "<identifier>" | ϵ
<topic_domain> := "<identifier>" | ϵ
<expert_name> := <expert_var> | ϵ
<detail_level> := <digit>{<digit>} | ϵ
<source_of_clause> :=
<source_var>=**any SourceOf**(<topic_var>, <detail_level>, <expert_name>)
<source_var> := $<identifier>
<detail_level> :={<digit>}
<a_condition_clause> := <topic_metalink_source_variable> <set_op> <built_in_function>
| <topic_metalink_source_variable> = <constant>
| <topic_metalink_source_variable> <set_op> <SQL-TC_query>
| <built_in_function> <op> <constant>
| <built_in_function> <set_op> <SQL-TC_query>
| <built_in_function>
| <constant> <set_op> <topic_metalink_source_variable>
<op> := = | < | <= | > | >=
<set_op> := <op> | **in** | =**any** | =**all**
<bool_not> := **not**
<constant> := <string_constant> | <numeric_constant>
<string_constant> := "{<digit> | <letter> | <underscore>}"
<numeric_constant> := <digit>{<digit>}[. | ,]{<digit>}
<order_by_clause>:= **order by** [<topic_var>] **importance**
(**stop after** {<digit>} **most important**
| **when importance below** <numeric_constant>
| **after** {<digit>} **most important and**
when importance below <numeric_constant>)

Appendix D

XTM Syntax

In the following, we briefly summarize and explain the elements and their attributes in [33] which is assumed to cover a functional and practical subset of [44] and also provide a convenient syntactical way of expressing the XTM Conceptual Model [32]:

- <topicMap> element: This is the root element that indicates the beginning of a topic map document and thus signals the syntactical recognition and conformance thereafter. It is a container for the fundamental members of a topic map -namely, <topic>, <association> and <mergeMap> elements.

- <topic> element: A <topic> element is a syntactical construct that represents a real-world subject in a computer system. The subject that is captured by the <topic> element may be either addressable or non- addressable [32]. Informally, an addressable subject is one that may be retrieved by a computer system and compared to other addressable subjects. For instance, a <topic> element about the electronic document of this thesis (e.g., thesis.pdf) is said to be capturing an addressable subject. On the other hand, a topic that captures the subject "United Nations Building" has a non-addressable subject; because the building itself may not be addressed in a computer system directly. However, there may be an information resource (which is addressable by definition) that embodies a human-interpretable

description of a non-addressable subject. Such a resource is called as a subject indicator. For instance, an electronic document that talks about the "United Nations Building" may be the subject indicating resource of the <topic> element in our example. These issues are discussed in [32, 31] in much more depth. A <topic> element in a topic map can itself be considered as a resource. Furthermore, each <topic> element is restricted to reify exactly one real-world subject. Each <topic> element has the required attribute id of type ID [34] which uniquely identifies a topic element in a topic map and provides a handle to reference a topic, as discussed below.

- <topicRef>, <subjectIndicatorRef> and <resourceRef> elements: As we have mentioned before, (almost) everything in a topic map is considered to be a topic. For instance, the type of a topic is another topic, the occurrence and association role types are topics and of course, the players of associations are topics. Clearly, this requires a way of referencing topics within one or more topic maps. The scheme supported in the current DTD employs the XLink [21] referencing mechanism through the <topicRef>, <subjectIndicatorRef> and <resourceRef> elements. Each of these elements provides a URI [2] reference through their *xlink:href* attribute. For the time being, all the links are *simple* as expressed with the fixed attribute *xlink:type*.

  In accordance with the discussion about the real-world subjects represented by a <topic> element, we may use one of these referencing elements: the <topicRef> element provides a URI reference asserting a link to a topic [32]. For the time being, this is achieved by the fragment identifier ("#") followed by the unique topic id in a topic map. The <subjectIndicatorRef> element provides a URI asserting a link to a resource which is an indicator of a particular (probably non-addressable) real-world subject. Thus, in effect, this element references the topic with this subject. Finally, the <resourceRef> element provides a URI reference asserting a link to a resource (which is, by definition, an addressable subject).

  Note that, all elements in this XTM DTD that reference another topic or resource make use of one or more of these elements as their sub-elements.

- <instanceOf> element: indicates class-instance relationship between the parent element including the <instanceOf> and the referenced <topic> element through either one of the <topicRef> or <subjectIndicatorRef> sub-elements. A <topic>, <association> or <occurrence> may include <instanceOf> as a sub-element.

- <subjectIdentity> element: This element may only be contained in a <topic> element and references the real-world subject captured by the parent <topic> element via the previously discussed referencing elements.

- <baseName> element: This element provides a base-name for the parent <topic> element through its <baseNameString> sub-element. The other name types (such as sort-name, display-name) declared in [44] are supported through a <variant> sub-element of <baseName> in the XTM DTD. Finally, the <scope> sub-element provides a context in which the assignment of the base-name to the parent <topic> is valid.

- <occurrence> element: This element can appear in only a <topic> element and provides a reference (link) to an information resource for the parent <topic> element. The resource may be referenced by <resourceRef> element. Alternatively, an <occurrence> element may have a <resourceData> sub-element that actually includes character data content giving information about the parent <topic>. The occurrence role type of [44] is supported through the <instanceOf> sub-element. The <scope> sub-element describes the scope in which the referenced information resources are valid for the parent <topic>.

- <association> element: expresses a relationship among a number of *member topics* that are expressed by <member> sub-element. The type of relationship is indicated via the <instanceOf> sub-element. An association may be valid in a certain scope which is again expressed using a <scope> child element.

- <member> element: A <member> element references player topics or resources in a parent <association> that play one and the same role in this <association>. The player elements are again referenced through one of the

mentioned referencing elements and the role played by them is indicated via the <roleSpec> child.

Note that an association may be a player in another association -just like any other topic- by referencing it through <subjectIndicatorRef>.

- <roleSpec> element: The role type itself is again a <topic>, and thus it is declared through either <topicRef> or <subjectIndicatorRef> child of <roleSpec> element.

- <scope> element: This element contains one or more references to topics, subject indicators and/or resources, the union of which defines a *context* of validity for a particular characteristic assignment. The <scope> element may be contained in <baseName>, <occurrence> and <association> elements to provide such a context.

- <mergeMap> element: A <mergeMap> element references an external topic map by means of the xlink:href attribute and causes that referenced topic map to be merged with the current (parent) topic map.

Appendix E

XTM for Expert Advice Repository

In the following, we provide a "pruned" version of an XTM document for a subset of advices of expert E1 in Appendix A.1. In this XTM, the following have declarative meaning in accordance with [32] and they have not been represented in the tabular forms of Appendix A.1.

- Top level (meta) topic types: horror novelist, novel, and movie.
- Top level (meta) topic and metalink domains: literature, cinema, and horror.
- Top level (meta) source (occurrence) roles: website, mentions, summary, and review.
- Top-level (meta) topic for metalink (association) types: writtenby, and based-upon.

The above (meta) topics should be declared because of the topic map motto "(almost) everything is a topic in a topic map". Although this paradigm provides a high level of expressive power for topic maps, it also makes them very long

and verbose. Also note that, the XTM document below includes some of the constructs introduced by us in Chapter 5.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE topicMap SYSTEM "xtm1.dtd">
<topicMap xmlns="http://www.topicmaps.org/xtm/1.0/"
          xmlns:xlink="http://www.w3.org/1999/xlink">

  <! - meta topics representing types ->

  <topic id="MT1">
    <basename>
      <basenameString> horror novelist</baseNameString>
    </basename>
  </topic>
  <topic id="MT2">
    <basename>
      <basenameString> novel</baseNameString>
    </basename>
  </topic>
  <topic id="MT3">
    <basename>
      <basenameString> movie</baseNameString>
    </basename>
  </topic>

  <!- meta topics representing domains ->

  <topic id="MT4">
    <basename>
      <basenameString> literature<basenameString>
    </basename>
  </topic>
```

```
<topic id="MT5">
   <basename>
      <basenameString> cinema</baseNameString>
   </basename>
</topic>
<topic id="MT6">
   <basename>
      <basenameString> horror</baseNameString>
   </basename>
</topic>

<!– meta topic representing
occurrence types (source role types) –>

<topic id="MT7">
   <basename>
      <basenameString> website</baseNameString>
   </basename>
</topic>
<topic id="MT8">
   <basename>
      <basenameString> mentions</baseNameString>
   </basename>
</topic>
<topic id="MT9">
   <basename>
      <basenameString> summary</baseNameString>
   </basename>
</topic>
<topic id="MT10">
   <basename>
      <basenameString> review</baseNameString>
   </basename>
```

```
</topic>

<! – meta topics representing association (metalink) types –>

<topic id="MT11">
   <basename>
      <basenameString> WrittenBy</baseNameString>
   </basename>
</topic>
<topic id="MT12">
   <basename>
      <basenameString> BasedUpon</baseNameString>
   </basename>
</topic>

<!– meta topics representing constructs defined by us –>

<topic id="MT13">
   <basename>
      <basenameString> antecedent</baseNameString>
   </basename>
</topic>
<topic id="MT14">
   <basename>
      <basenameString> consequent</baseNameString>
   </basename>
</topic>
<topic id="MT15">
   <basename>
      <basenameString> HasTAdvice</baseNameString>
   </basename>
</topic>
<topic id="MT16">
```

```
    <basename>
      <basenameString> HasMAdvice</baseNameString>
    </basename>
  </topic>
  <topic id="MT17">
    <basename>
      <basenameString> HasSAdvice </baseNameString>
    </basename>
  </topic>
  <topic id="MT18">
    <basename>
      <basenameString> 1 </baseNameString>
    </basename>
  </topic>
  <topic id="MT19">
    <basename>
      <basenameString> 0.8</baseNameString>
    </basename>
  </topic>
  <topic id="MT20">
    <basename>
      <basenameString> 0.7 </baseNameString>
    </basename>
  </topic>
  <topic id="MT21">
    <basename>
      <basenameString> 0.6 </baseNameString>
    </basename>
  </topic>
  <topic id="MT22">
    <basename>
      <basenameString> 0.5 </baseNameString>
    </basename>
```

```
</topic>
<topic id="MT23">
   <basename>
      <basenameString> hasSDetailLevel </baseNameString>
   </basename>
</topic>
<topic id="MT24">
   <basename>
      <basenameString> 4 </baseNameString>
   </basename>
</topic>
<topic id="MT25">
   <basename>
      <basenameString> source </baseNameString>
   </basename>
</topic>
<topic id="MT26">
   <basename>
      <basenameString> source detail level </baseNameString>
   </basename>
</topic>

<!- topic representing novelist Stephen King in domain literature->

<topic id="T1">
   <instanceOf>
      <topicRef xlink:href="#MT1"/>
   </instanceOf>
   <subjectIdentity>
      <subjectIndicatorRef xlink:href="http://www.stephenking.com"/>
   </subjectIdentity>
   <baseName>
      <baseNameString> Stephen King</baseNameString>
```

```
      <scope>
         <topicRef xlink:href="#MT4"/>
      </scope>
   </baseName>
   <occurrence id="S1">
      <instanceOf>
         <topicRef xlink:href="#MT7"/>
      </instanceOf>
      <resourceRef xlink:href="http://www.king.com"/>
   </occurrence>
   <occurrence id="S2">
      <instanceOf>
         <topicRef xlink:href="#MT8"/>
      </instanceOf>
      <resourceRef xlink:href=
         "http://www.newsweek.com/17-10-2000/tragic-accident.html "/>
   </occurrence>
</topic>

<!- topic representing the novel Carrie in domain literature ->

<topic id="T2">
   <instanceOf>
      <topicRef xlink:href="#MT2"/>
   </instanceOf>
   <baseName>
      <baseNameString> Carrie</baseNameString>
      <scope>
         <topicRef xlink:href="#MT4"/>
      </scope>
   </baseName>
   <occurrence id="S3">
      <instanceOf>
```

```
            <topicRef xlink:href="#MT9"/>
        </instanceOf>
        <resourceRef xlink:href="http://www.books.com/carrie.html"/>
    </occurrence>
    <occurrence id="S4">
        <instanceOf>
            <topicRef xlink:href="#MT10"/>
        </instanceOf>
        <resourceRef xlink:href="http://www.critics.com/carrie.html "/>
    </occurrence>
</topic>

<!– topic representing the movie Story of Carrie in cinema domain –>

<topic id="T4">
    <instanceOf>
        <topicRef xlink:href="#MT3"/>
    </instanceOf>
    <baseName>
        <baseNameString> Story of Carrie</baseNameString>
        <scope>
            <topicRef xlink:href="#MT5"/>
        </scope>
            </baseName>
</topic>

<!– association representing metalink instance
"Stephen King"→ WrittenBy "Carrie" –>

<association id="M1">
    <instanceOf>
        <topicRef xlink:href="#MT11"/>
    </instanceOf>
```

```
    <scope>
      <topicRef xlink:href="#MT4"/>
      <topicRef xlink:href="#MT6"/>
    </scope>
    <member>
      <roleSpec>
        <topicRef xlink:href="#MT13"/>
      </roleSpec>
      <topicRef xlink:href="#T1"/>
    </member>
    <member>
      <roleSpec>
        <topicRef xlink:href="#MT14"/>
      </roleSpec>
      <topicRef xlink:href="#T2"/>
    </member>
</association>

<!– association representing metalink instance
"Carrie"→BasedUpon "Story Of Carrie" –>

<association id="M7">
    <instanceOf>
      <topicRef xlink:href="#MT12"/>
    </instanceOf>
    <member>
      <roleSpec>
        <topicRef xlink:href="#MT13"/>
      </roleSpec>
      <topicRef xlink:href="#T2"/>
    </member>
    <member>
      <roleSpec>
```

```
            <topicRef xlink:href="#MT14"/>
        </roleSpec>
        <topicRef xlink:href="#T4"/>
    </member>
</association>

<!– the following are (meta) associations used to express
 importance values attached to topics, metalink instances and source. –>

<association id="MM1">
    <instanceOf>
        <topicRef xlink:href="#MT15"/>
    </instanceOf>
    <member>
        <roleSpec>
            <topicRef xlink:href="#MT13"/>
        </roleSpec>
        <topicRef xlink:href="#MT18"/>
    </member>
    <member>
        <roleSpec>
            <topicRef xlink:href="#MT14"/>
        </roleSpec>
        <topicRef xlink:href="#T1"/>
    </member>
</association>

<association id="MM2">
    <instanceOf>
        <topicRef xlink:href="#MT15"/>
    </instanceOf>
    <member>
        <roleSpec>
```

```
            <topicRef xlink:href="#MT13"/>
        </roleSpec>
        <topicRef xlink:href="#MT18"/>
    </member>
    <member>
        <roleSpec>
            <topicRef xlink:href="#MT14"/>
        </roleSpec>
        <topicRef xlink:href="#T2"/>
    </member>
</association>

<association id="MM3">
    <instanceOf>
        <topicRef xlink:href="#MT15"/>
    </instanceOf>
    <member>
        <roleSpec>
            <topicRef xlink:href="#MT13"/>
        </roleSpec>
        <topicRef xlink:href="#MT20"/>
    </member>
    <member>
        <roleSpec>
            <topicRef xlink:href="#MT14"/>
        </roleSpec>
        <topicRef xlink:href="#T4"/>
    </member>
</association>

<association id="MM4">
    <instanceOf>
        <topicRef xlink:href="#MT16"/>
```

```
    </instanceOf>
    <member>
      <roleSpec>
        <topicRef xlink:href="#MT13"/>
      </roleSpec>
      <topicRef xlink:href="#MT18"/>
    </member>
    <member>
      <roleSpec>
        <topicRef xlink:href="#MT14"/>
      </roleSpec>
      <subjectIndicatorRef
          xlink:href="http://www.sql-tc.com/king.xtm#M1"/>
    </member>
  </association>

  <association id="MM5">
    <instanceOf>
      <topicRef xlink:href="#MT16"/>
    </instanceOf>
    <member>
      <roleSpec>
        <topicRef xlink:href="#MT13"/>
      </roleSpec>
      <topicRef xlink:href="#MT21"/>
    </member>
    <member>
      <roleSpec>
        <topicRef xlink:href="#MT14"/>
      </roleSpec>
      <subjectIndicatorRef
          xlink:href="http://www.sql-tc.com/king.xtm #M7"/>
    </member>
```

```
</association>

<association id="MM6">
   <instanceOf>
      <topicRef xlink:href="#MT17"/>
   </instanceOf>
   <member>
      <roleSpec>
         <topicRef xlink:href="#MT13"/>
      </roleSpec>
      <topicRef xlink:href="#MT18"/>
   </member>
   <member>
      <roleSpec>
         <topicRef xlink:href="#MT14"/>
      </roleSpec>
      <subjectIndicatorRef
         xlink:href="http://www.sql-tc.com/king.xtm#S1"/>
   </member>
</association>

<association id="MM7">
   <instanceOf>
      <topicRef xlink:href="#MT17"/>
   </instanceOf>
   <member>
      <roleSpec>
         <topicRef xlink:href="#MT13"/>
      </roleSpec>
      <topicRef xlink:href="#MT20"/>
   </member>
   <member>
      <roleSpec>
```

```
            <topicRef xlink:href="#MT14"/>
        </roleSpec>
        <subjectIndicatorRef
            xlink:href="http://www.sql-tc.com/king.xtm#S2"/>
    </member>
</association>

<association id="MM8">
    <instanceOf>
        <topicRef xlink:href="#MT17"/>
    </instanceOf>
    <member>
        <roleSpec>
            <topicRef xlink:href="#MT13"/>
        </roleSpec>
        <topicRef xlink:href="#MT22"/>
    </member>
    <member>
        <roleSpec>
            <topicRef xlink:href="#MT14"/>
        </roleSpec>
        <subjectIndicatorRef
            xlink:href="http://www.sql-tc.com/king.xtm#S3"/>
    </member>
</association>

<association id="MM9">
    <instanceOf>
        <topicRef xlink:href="#MT17"/>
    </instanceOf>
    <member>
        <roleSpec>
            <topicRef xlink:href="#MT13"/>
```

```
        </roleSpec>
        <topicRef xlink:href="#MT19"/>
      </member>
      <member>
        <roleSpec>
          <topicRef xlink:href="#MT14"/>
        </roleSpec>
        <subjectIndicatorRef
          xlink:href="http://www.sql-tc.com/king.xtm #S4"/>
      </member>
    </association>

    <!- the following is a (meta) association used to express
     that source (occurrence) S4 has the detail level 4 ->
    <association id="MM10">
      <instanceOf>
        <topicRef xlink:href="#MT23"/>
      </instanceOf>
      <member>
        <roleSpec>
          <topicRef xlink:href="#MT26"/>
        </roleSpec>
        <topicRef xlink:href="#MT24"/>
      </member>
      <member>
        <roleSpec>
          <topicRef xlink:href="#M25"/>
        </roleSpec>
        <subjectIndicatorRef
          xlink:href="http://www.sql-tc.com/king.xtm#S4"/>
      </member>
    </association>
</topicMap>
```