Database Querying on the World Wide Web

Carlos F. Enguix, Joseph G. Davis, and Aditya K. Ghose

Technical Report 1998/5/101 May 1998 (Updated Version 1999)

DECISION SYSTEMS LABORATORY

Department of Business Systems

The University of Wollongong

Northfields Ave.

Wollongong NSW 2522, Australia

cenguix@acm.org, joseph_davis@uow.edu.au , aditya@uow.edu.au

Abstract

The World Wide Web can be considered to be a very large semi-structured database that holds vast amounts of useful information. However, our ability to query, search, and reuse the information on the Web is limited at present. Existing search techniques suffer from critical deficiencies with respect to robustness, flexibility, and precision. This research attempts to develop a domain-centred alternative to keyword and subject directory search engines. By focusing on a well-demarcated (logical) domain whose ontology can be modelled using third-generation, object-relational data models, we show how the relevant Web data can be restructured using an object-relational database, against which SQL-type queries can be issued. A prototype implementation for the 'universities' domain entitled 'UniGuide' is presented.

1. Introduction

The potential of the Internet and the World Wide Web (WWW or the Web) for enabling the development of complex, distributed information systems applications is well known. The Web can be viewed as a very large semi-structured database which holds vast amounts of useful information. However, the ability of users and applications to use and reuse this information is severely restricted at present due to factors such as:

- The explosion in the quantum of data available on the Web. This renders the problem of discovering in reasonable time the information resources required by users, somewhat difficult. The rate of growth of resources on the Web has been proceeding at an exponential rate due to the ever increasing number of documents, publications, and other information being placed on it by a growing number of users. A recent estimate puts the total number of pages on the Web at around 200 million and growing at the rate of 200% per year [Bharat and Broder 1998].
- Diversity in the information placed on the Web and the diversity of the user community
- The limitations of the current (first generation) Web browsing and search tools. The efficacy and usefulness of the existing keyword and subject directory-based search engines has been under severe strain as evidenced by well-documented problems such as:

    a. imprecise and inefficient search
    b. inability to retrieve fragments of useful information
    c. production of a long list of useless documents [Poulter 1997]

Vercoustre has described the current situation on the Web as being characterised by a focus on pushing more and more information on the Web. As a consequence, it is claimed that already more data exists on intranets than there is on all relational databases combined [Vercoustre 1998]. In the light of this, it is critical that our ability to search, query and use the data on the Web can improve beyond capabilities and features offered by the current generation of search engines. The enormity of the problem of searching, retrieving and using Web data suggests the need for alternative metaphors and conceptualisations that go beyond keyword-based searches. We propose an alternative approach to the problem that views the Web as a database and facilitates database querying of Web data, albeit within restricted logical domains such as universities, health care sector, book stores, software industry, etc. We address some of the conceptual and practical questions dealing with developing and structuring ontologies (models of concepts and their interrelationships) within such well-demarcated domains. The ontology model is structured as an object-relational database schema which is the basis for the development of an object-relational database that stores structured data extracted from the meta-tags associated with relevant Web pages in the chosen domain. SQL3 queries can be issued against this database. A prototype implementation of such a database query search engine entitled *UniGuide* for the domain of "Australian universities" is presented. Methodological guidelines for the design and implementation of similar domain-focused database search engines are outlined.

The paper is organised as follows: in section 2 we present an overview of the relevant literature. A model that captures the core constructs and their interrelationships (ontology model) for the 'Australian Universities' domain and the architecture and implementation of the *UniGuide* prototype are presented in section 3. The usage of the prototype from an end-user perspective is outlined in section 4. We conclude this paper with section 5, including a list of conclusions.

2.Overview of the relevant literature.

2.1. Structuring the Web Data

The World Wide Web was developed with the goal of pooling human knowledge so that collaborators in remote sites can share ideas and information on a common project [Berners-Lee et al. 1994]. It has grown into a large global, distributed, heterogeneous collection of documents connected by hyperlinks. Much of the data on the Web (typically in the form of HTML documents) can best be described as semi-structured, given that its structure is often implicit, and not strictly typed or regular as that found in standard database systems [Abiteboul 1997]. Extracting the structure of an individual HTML document or a collection of such documents is a challenging problem in view of the absence of a predefined standard or schema. Often the schema can only be derived post hoc, after the existence of data as compared to conventional databases, even if the schema can be sometimes very large and constantly evolving. Despite the difficulties involved, Abiteboul emphasizes the need for building a more structured layer on top of an irregular and less controlled layer of documents and files so as to achieve "... important gains in answering the most common queries" [Abiteboul 1997]. This structured layer can be populated with information extracted from the semi-structured lower layer. There is a general consensus that this higher structured layer can offer a flexible and efficient

access to the information in the lower layer and thereby provide the benefits of standard database access methods [Abiteboul 1997; Hammer et al. 1997].

In a similar vein, Han et al. (1995) have proposed a multi-layered database (MLDB) with the primary motivation of discovering resources and knowledge on the internet. In their scheme, layer 0 is the whole internet (the primitive information in the global information base) while the higher layers store generalised information and transformations of the information that resides in layer 0. They also propose that layer1 and higher layers can be modelled using an extended-relational or object-oriented data model. Information at all the layers barring the primitive layer 0 can be managed by database technology, thus facilitating efficient and controlled search for knowledge discovery. The MLDB approach has great conceptual appeal but the significant practical problems in implementing such systems have been inadequately addressed in Han et al. (1995).

One of the more challenging questions in creating such structured layers relate to the source, mechanisms and methods for extracting relevant information from the semi-structured layer (WWW) to the structured layers above. While it is possible to develop wrapper programs for individual Web sites that can extract relevant data into relational database structures [Rajamaran 1998], an alternative approach is that attaching metadata that describe the contents of individual Web pages. This would permit us to view pertinent information about Web pages as a series of structured tuples of data. This approach is partly based on the assumption that metadata will/should be treated as first class objects [W3C 1998a] and will serve as the interface from the WWW to a structured database. Because of the exclusive focus on metadata, there is no need for strict typing over the contents of the HTML documents but only over the metadata. It is worth noting that a necessary condition for the transformation of the Web into a database is the integration of critical database features such as metadata into the Web.

The type of metadata to be attached to Web pages is a series of customised metatags that describe entity instances represented by the contents of target Web pages. Examples of the more common use of metatags include: keywords and description used by some of the most popular search engines. Another important standard is the one for metatags proposed by the Dublin Core, a 15-element metadata set intended to facilitate discovery of electronic resources [OCLC 1997]. New standards such as XML, XML-Schemas and Resource Description Framework (RDF) are emerging on the Web in order to improve the access and retrieval of Web resources. XML, XML-Schemas and RDF facilitate the definition of new vocabularies specific to certain communities, which is likely to facilitate the automatic extraction of metadata specific to these communities [W3C 1998b].

2.2 The Query Problem: Deficiencies in Current Search Engines

A majority of the existing search engines provide a very simple interface to querying, a simple text box for entering keywords. We list below some of the more common deficiencies of current implementations:

- Most of the search engines are keyword-based [Poulter 1997], constrained to very limited structured querying thus providing more syntactic and less semantic precision
- The lack of control in querying Web data: the boundaries of the query are unknown and the output of a query is hard to predict
- The ability to establish associations between data elements is scarce or non-existent

Many of the above problems can be traced to the absence of a conceptual model that can cover the whole semantics of the WWW. Keyword-based search engines rely on the use of huge indexes that are mapped to URLs. Their power typically resides in efficient statistical algorithms for matching keywords with the contents of Web pages (Yuwono and Lee 1996). The lack of semantics leads to the situation in which it is almost impossible to establish relationships or logical associations between concepts or entities. Depending on the internal algorithms of the different search engines we can expect different results in response to queries. Also, since the search is based on syntactic precision, we usually end up with a very large number of unrelated hits which makes it difficult to find the required information.

Table 1 summarises the main differences between keyword-based search engines and database querying on the WWW.

Properties	Classification			
	Keyword-based		**Database querying**	
Query Composition	> + Flexible, + Easy < + Vague, Diffuse		> + Precise, + Constrained < + Complex	
Query Precision	< + Lexical, Syntactical levels		> + Semantic level	
Data Relations	< + Scarce/non-existent, difficult		> + Feasible, abundant	
Output of results	< + Unpredictable		> + Predictable	
Data Structures	> - Infrastructure: Hypertext + Indexes		< + Infrastructure: Schema + Structured Data + Indexes	
Data typing	< + Non-typed/loosely typed		> + Strictly typed	
Applicability	> Universal/Generic		< Domain-specific	
Legend	> Strength	< Weakness	+ More	- Less

Table 1: Keyword-based Search Engines vs Database Querying on the WWW.

2.3 Related Research.

Atzeni et al. (1997) have proposed a data model and a view definition language to represent, restructure, and query the information stored in structured Web servers, i.e. those servers and Web sites in which data is organised according to relatively precise structures and pages present strong regularities. The focus is on exploiting the degree of structure (the hypertextual structure and the textual organisation reflected in the HTML tags) to facilitate the extraction of attribute values using a text restructuring language to build relational views of Web data on specific servers. These views can be queried using a relational query language [Atzeni et al. 1997].

Rajaraman has outlined the virtual database technology (VDB) that attempts to make the Web (and other external data sources) behave as an extension of an enterprise relational database system. VDB is designed to enable the gathering, structuring, and integration of data from disparate data sources including Web pages and provides the user or application programmer with the appearance of a single unified relational database, which can be queried. The implementation strategy is essentially bottom-up and is based on extracting and integrating relevant data into a relational database using separate wrapper programs containing appropriate extraction rules for each Web site (Rajaraman 1998). A similar approach is reported in Ashish and Knoblock (1997). The architecture of a system for querying Web data using a Query-By-Example (QBE) or SQL interface is presented in Sunderraman (1997). A conversion module converts Web data in HTML format into relational data format. In addition, a methodology for searching and querying Web data which have some degree or relational structure is included [Sunderraman 1997].

These are emerging indications that given the sheer scale of the Web, any attempt to develop a model to represent the semantics of the entire Web is unlikely to yield useful results. This, in our view, is the critical problem with the MLDB approach presented by [Han et al. 1995]. As well, some of the more recent research is pointing in the direction of the need to focus more realistically on restricted logical domains whose ontologies can be explicated and modelled. This strategy of "divide and conquer" involves identifying and isolating reasonably well-demarcated subsets of the Web consisting of a collection of homogeneous Web sites whose structure can be modelled. This implies the existence of generic entities or concepts in the domain and the stable and predictable inter-relationships among them. One should not lose sight of the fact that any modelling is a process of abstraction. Hence it is impossible to model all the different variations of a given entity or to incorporate all possible entities in a "characteristic" domain. In view of the similarity in structure that exists within the domain, it is our contention that all significant and common entities and their inter-relationship can be represented by our approach which is consistent with the observation by Atzeni et al. (1997) that in structured servers and intranet applications, the hypertext organisation of the Web pages tend to mirror the underlying structure of the organisation (or domain).

3. *UniGuide*: Architecture and Implementation

3.1 Introduction

The framework proposed in this paper is predicated on two significant assumptions:

- Ontologies or models of concepts and their relationships [Mahalingam and Huhns 1997] represent powerful means to structure the global information base on the Web
- The range and diversity of data on the Web is so extensive that ontologies may have to be constructed separately for each relatively well defined domain. Further the structure of a collection of Web documents that fall within the purview of a particular domain can be modelled as an ontology which in turn can guide Web search.

Ontology is a term with a long pedigree in philosophy. It refers to things that 'exist' (in the domain). It can be thought of as a generic description of the concepts and relationships that always exist, enabling knowledge sharing and reuse. For instance, it is reasonable to expect that the university domain will always have information regarding research entities, academic departments, courses, research outputs, and so on. Furthermore, these are likely to be inter-related in similar and predictable ways. As well, ontologies can grow and shrink based on the context in which they are used [Mahalingam and Huhns 1997].

Our proposed method involves isolating a distinct domain, modelling its ontology using an object-relational data model, and extracting and storing all relevant metadata from the domain Web pages in database tables corresponding to the objects in the model. This database becomes a resource that can be queried by end-users for a wide range of information specific to the domain in a fashion that current search engines cannot match. Populating the database can be automated using suitable indexing robots.

UniGuide is a demonstration prototype implementation of the above approach. The ontology for all Australian Universities Web pages is modelled as an object-relational data model. This model is then mapped to an object-relational database and a set of queries that can be issued against this database is presented in subsequent sections.

3.2 The Object-Relational Data Model

The ontology could be represented as a pure relational model but the complexity of objects in the domain and the hierarchical structure of the university domain suggests the need for object-oriented approaches. Hierarchical structures can be represented in a more natural way in an object-oriented paradigm. However, a pure object oriented model is not the most suitable one if we want to keep the model compact, parsimonious, and easy to understand, without having to depict all subclasses or subtypes or a given object class. A hybrid,

third-generation, object-relational data model meets most of the requirements incorporating aspects of relations (tables) and complex objects while also mapping directly to an object-relational database system.

The object-relational model for *UniGuide* is shown in figure-1. It shows the entities/objects that have been modelled currently but the model is extensible. The objects in the model represent the Web pages of corresponding entities in the university Web sites. There is a 1:M relation (R[1:M]) between these objects and URLs. Generally, a Web page may contain many entity-instances but an entity-instance is associated to one and only one URL. In the UniGuide ORDB, we can represent traditional relational constructs such as tables and rows as well as object-oriented constructs such as object identifiers and sets.

A prototype row instance contains the following attributes:

- **oid:** Object Identifier (physical identifier of an entity instance)
- **URL:** Hypertext Link to resource described, generally homepage
- **Components Primary Key:** logical identifiers of an entity instance (uniqueness constraint)
- **Other Functional Dependent Attributes:** other descriptive information
- **Last_Modified:** Timestamp attribute in order to guide the indexing robot if a particular entity instance should be updated or not.

A prototype set instance contains the following attributes:

- **oid:** Object Identifier (physical identifier of a set instance)
- **URL:** Hypertext Link to resource described, generally homepage

- **Components Pseudo Primary Key:** rules-based virtual primary key in unordered multi-set instances
- **Other Functional Dependent Attributes:** other descriptive information
- **Last_Modified:** Timestamp attribute in order to guide the indexing robot if a particular entity instance should be updated or not.
- **poid:** Object identifier of the parent container object. Relates container and subordinate (aggregate) objects

The object-relational representation of the ontology model for Australian Universities is shown in figure1. An explanation of the entity types in the model is provided below:

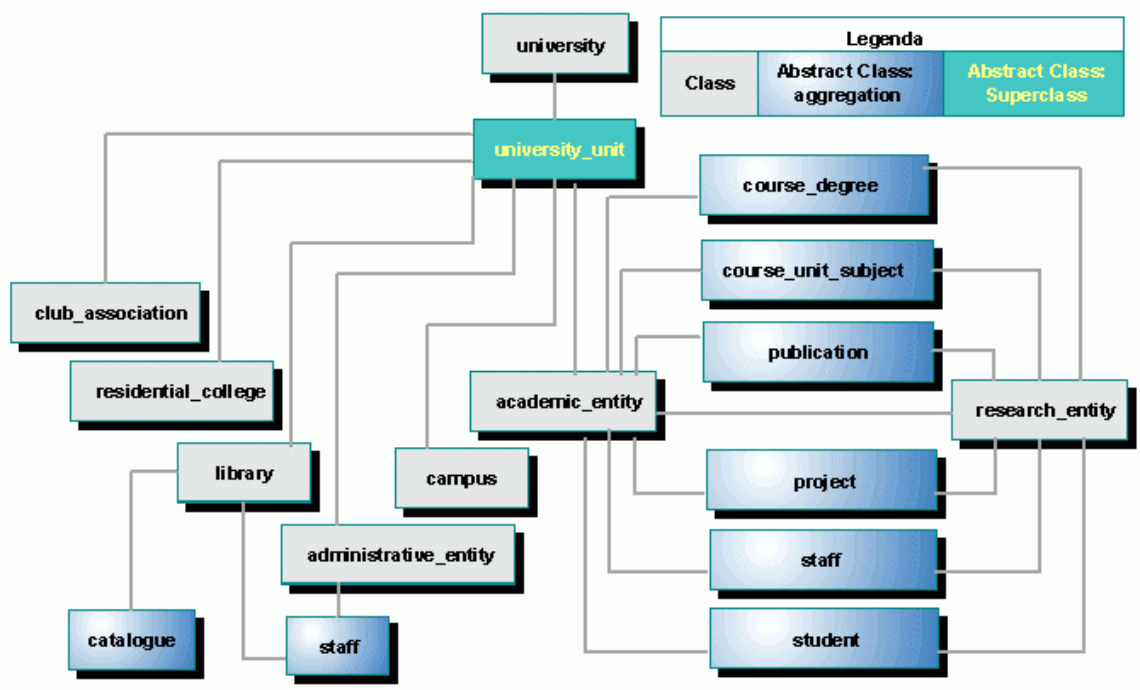

Figure 1. Synthesised graphical representation of the Object Relational Model of *UniGuide*

- **Entity classes:** autonomous and individual classes. Includes, universities, administrative entity (division, office, etc.), academic entities (faculties, departments, schools, etc.), library, etc
- **Abstract superclasses:** used to express type/subtype specialisation. Includes University unit which is the superclass of a majority of entities: administrative entity (division, office, etc.), academic entities (faculties, departments, schools, etc.) and so on.
- **Abstract aggregate classes:** represents containment hierarchies and aggregation. For instance, an academic entity may contain a set of staff, courses, subjects, etc.
- **Association classes:** models relationships between entity classes. Includes for instance, academic-entity/research-entity, research-entity-university, etc

3.3 Components of *UniGuide*: Overview

A World Wide Web search engine is defined as a retrieval service, consisting basically of a harvester (indexing robot), a database, search software and a user interface available via WWW [Poulter 1997]. *UniGuide* has similar components with some subtle differences. The database is an object-relational hybrid, with the capability to handle sets, arrays, abstract data types (ADTs), object identifiers, references, relations, user defined and dynamically loadable executable functions, inheritance, rules, etc. [Stonebraker and Moore 1996]. Our implementation is carried out using ILLUSTRA ORDBMS®, which is an extended relational database system. Includes a computationally complete declarative query language (SQL3) capable of defining abstract and constructed data types and a powerful event-oriented rules sub-system.

The database search engine is implemented in a client-server environment and is based on the following core components:

- **The ORDB:** stores the University domain entities, the auxiliary catalogue system, the complex rules defined in the rules sub-system and Web applications as hybrid object-relational database constructs

- **The Auxiliary Catalogue System:** this component is required by the SQL3 Generator to generate SQL3 DML and SELECT statements from data captured from either forms-based manual input or metadata retrieved by the indexing agent. Stores metadata about tables, sets, attributes and so on.

- **The Rules Subsystem:** an event-driven generic rules subsystem capable of handling traditional integrity constraints, referential integrity, etc.

- **The SQL3 Generator:** a library of dynamically loadable executable external modules capable of generating SQL3 statements "on the fly" which are executed a posteriori.

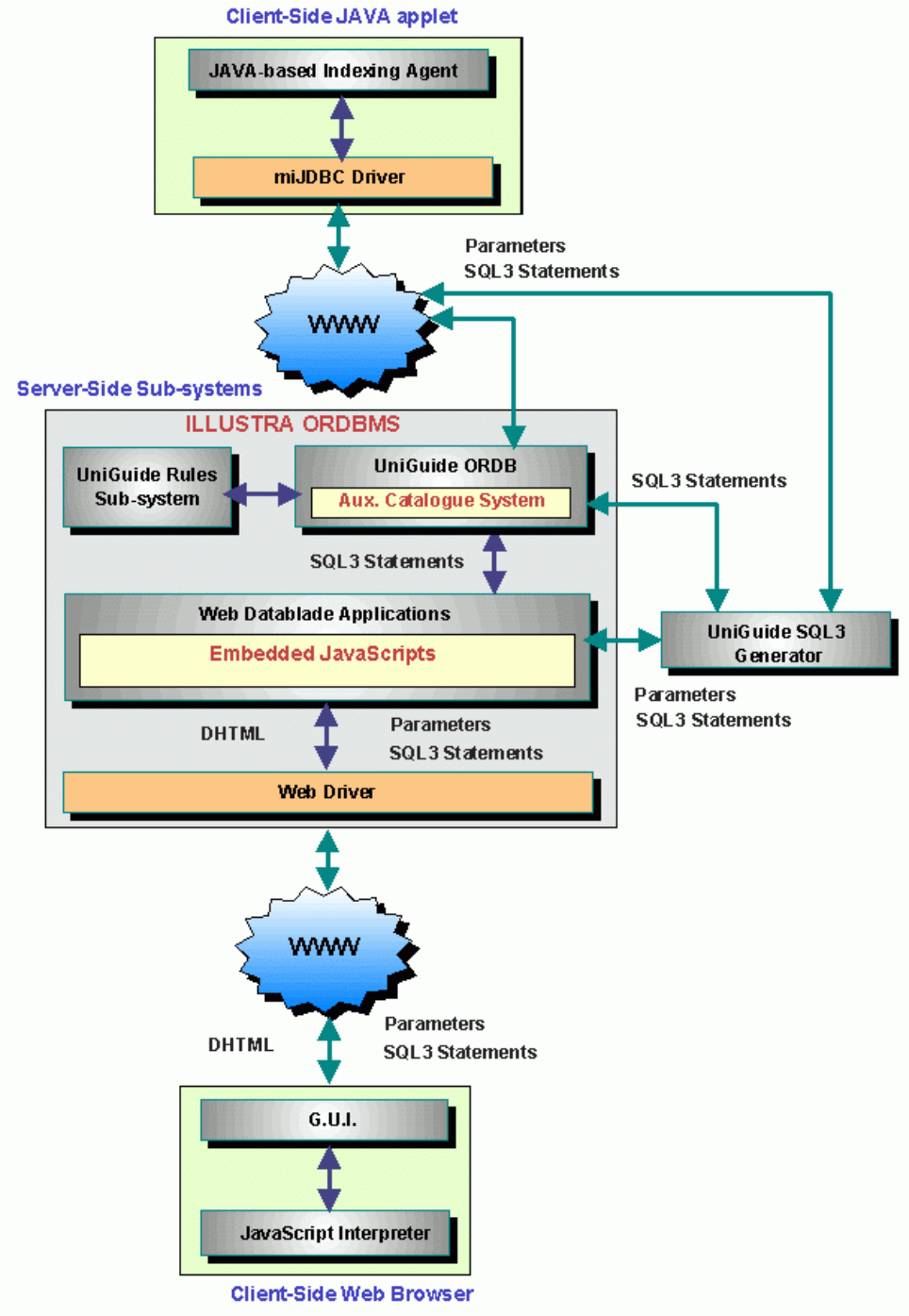

Figure 2. High-level overview of UniGuide components

The search software is implemented in a client-server environment and is based on SQL3 queries which can call to dynamically loadable external user-defined functions (SQL3 Generator) with the ability to run other SQL queries as well ("callback" feature) and rules. The system currently comprises a respectable number of encapsulated rules used intensively in order to support rules-based check constraints, rules-based case-insensitive referential integrity, uniqueness of sets, automatic actualisation of object references, timestamp values and hypertext links. The definition of rules-based integrity constraints and rules-based referential integrity rather than defining them declaratively with SQL3 DDL statements (i.e. CHECK & REFERENCES clause) is more suitable for a loosely typed and dynamic Web environment. More flexible rules-based integrity constraints such as fuzzy referential integrity may be incorporated or redefined at any moment in order to adapt to new situations, without affecting the static structure of a table. The interface is generated dynamically in HTML by ILLUSTRA Web Datablade® applications and Javascripts, enabling any Javascript® compatible browser to access *UniGuide*. Figure 2 outlines the main components of the *UniGuide* prototype.

3.4 Meta-tags

Our proposal places heavily reliance on the availability of meta-tags in Webpages. These meta-tags represent metadata associated with target Webpages that correspond to core entities included in our model (i.e. academic entities, publications, projects, etc.). The *UniGuide* indexing robot scans all the Web pages in the Australian Universities domain in order to extract the structured meta-tags stored with the contents of the target Webpages. Figure 3 shows diagrammatically how metadata is extracted into the *UniGuide* database from different Web sites (such as University of Sydney, Monash University, etc.). From our viewpoint Webpages are containers of text that hold semi-structured data (HTML tags and text) and structured data (meta-tags). The core information of a given Web page from *UniGuide* perspective is not the Webpage itself but the metadata stored with the contents of the Webpage. The *UniGuide* indexing robot localises the required meta-tags and extracts their contents; it ignores the rest of the text and continues navigating through the links to other Web pages.

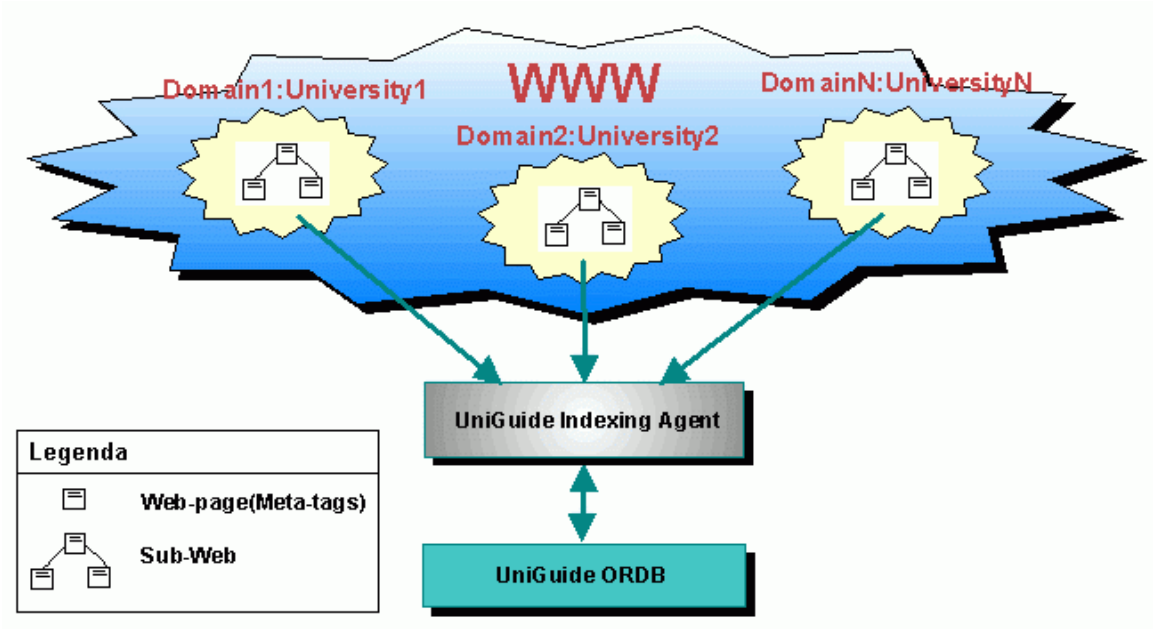

Figure 3. Harvest of meta-tags in *UniGuide*

Figure 4 presents an illustrative meta-tag representing an entity-instance of an academic entity/course_degree:

<!-- mandatory columns marked with * -->

```
<!-- Please Enter Values inside ' ' -->
<meta name="academic_entity_course/degree" scheme="UniGuide"
content="
(~ uni_id       [*university]= 'University of Technology Sydney' ~),
(~ academic_entity_type [*academic entity type]= 'Department' ~),
(~ academic_entity_name [*academic entity name]= 'Computer Science' ~),
(~ course_name  [*course name]= 'Bachelor of Science' ~),
(~ course_spec  [course speciality]= 'Computing Science' ~),
(~ course_type  [course type]= 'Undergraduate' ~),
(~ course_degree_type   [course degree type]= 'Single' ~),
(~ course_semesters [course semesters]= '6' ~),
(~ course_credits [course credits]= '144' ~),
(~ course_desc  [course description]= 'This course aims to provide a
sound education in all aspects of computing for students who intend to
make a career in the profession' ~)">
```

Figure 4. Examples of *UniGuide* Scheme meta-tags

The scheme attribute identifies the type of meta-tag represented, in this case a *UniGuide* scheme meta-tag. These meta-tags ideally must be inserted in the header of target Web pages in order to enable the *UniGuide* indexing agent to extract the information in an efficient manner.

3.5 UniGuide Indexing Agent

In the *UniGuide* indexing agent full-text indexing is ignored. The objective is to detect, parse and store in a breadth-first manner the information contained in *UniGuide* scheme metadata. The prototype scans only Web sites that belong to University domains, ignoring links to external Web sites. We have conducted a respectful number of experiments, traversing University domains that included the order of hundreds of thousands of URL's and in restricted sub-Webs that contained the required metadata. The following list includes the core components in the implementation of the prototype:

- **Indexing Agent Coordinator:** main application class that coordinates and monitors the indexing process in a breadth-first and multi-threaded manner
- **Indexing Statistics Manager:** stores general indexing agent statistics such as number of URLs visited, number of errors generated, number of *UniGuide* scheme meta tags processed and so on
- **URL Scanners:** the coordinator spawns individual URL Scanner threads that scan and extract URLs and *UniGuide* scheme metadata referenced in the contents of Web pages. When the required metadata is detected, the URL Scanner invokes the UniGuide SQL3 Generator, passing parameters and metadata via a JDBC driver. The generated SQL3 DML statement is executed next.
- **Thread Limiter:** used to limit the number of concurrent URL Scanner threads executing simultaneously. Based on the implementation of semaphores
- **Stop URL Scanner Daemon:** used to stop URL Scanner threads that have exceeded the maximum allowed time.

The following figure (fig. 5) shows the GUI of the *UniGuide* Indexing Agent (old version). The Meta Data section displays meta data statistics, which includes the current number of meta tags being processed, the last entity instance type detected, and the status code returned after processing the meta tag (ignored, inserted, updated or error).

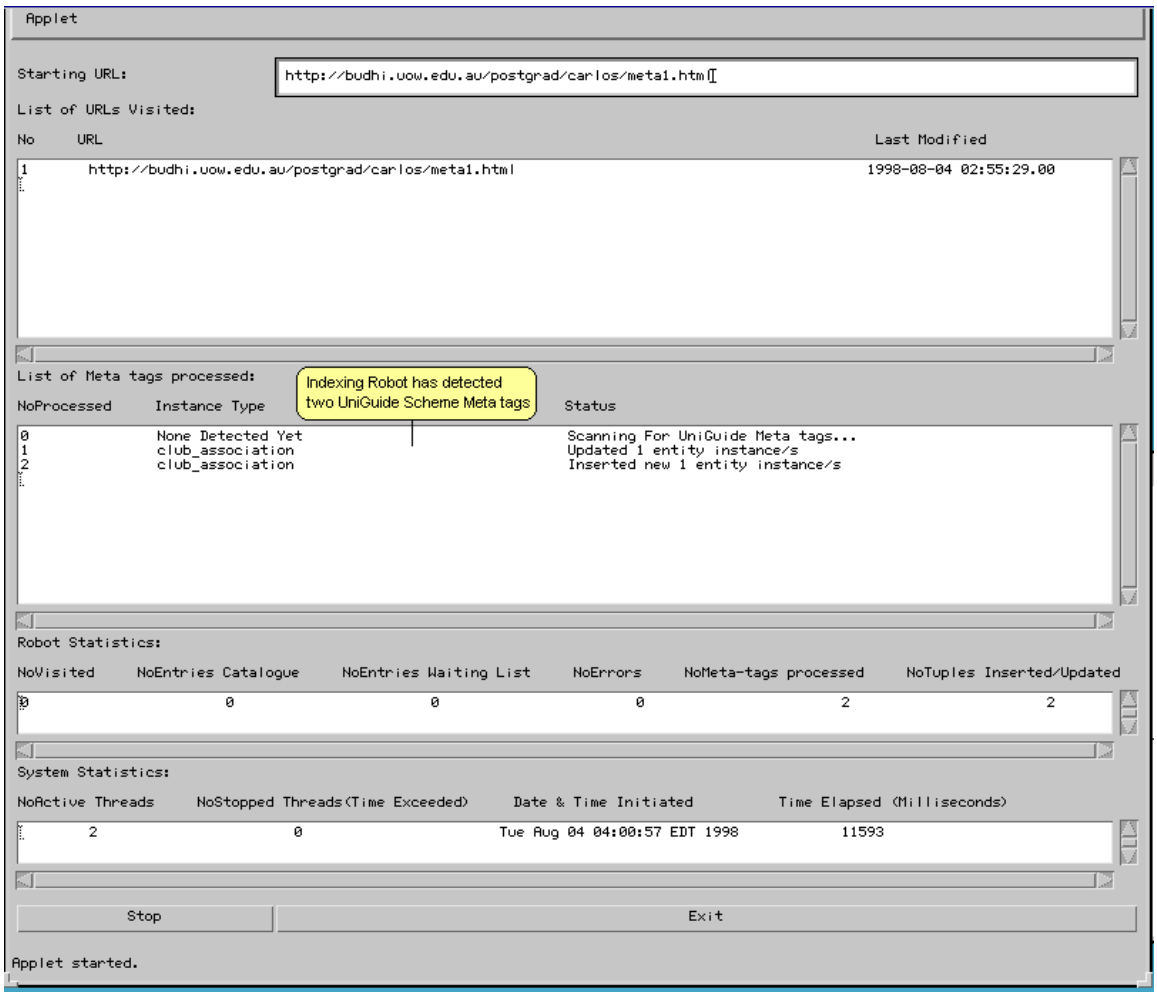

Figure 5. Indexing Agent Detecting UniGuide Scheme Metadata

4. *UniGuide*: End-User Perspective

4.1 Components

There are three distinct sections in *UniGuide* from an end-user perspective:

1. Submit URL, which allows the user to manually populate the database (see Fig. 6)
2. Queries, which allows the definition of simple and complex SQL3 queries and
3. The Meta Tag Generator, that generates *UniGuide* Scheme meta-data.

We shall describe only the query section, which includes the Simple Query section and the Free SQL query section.

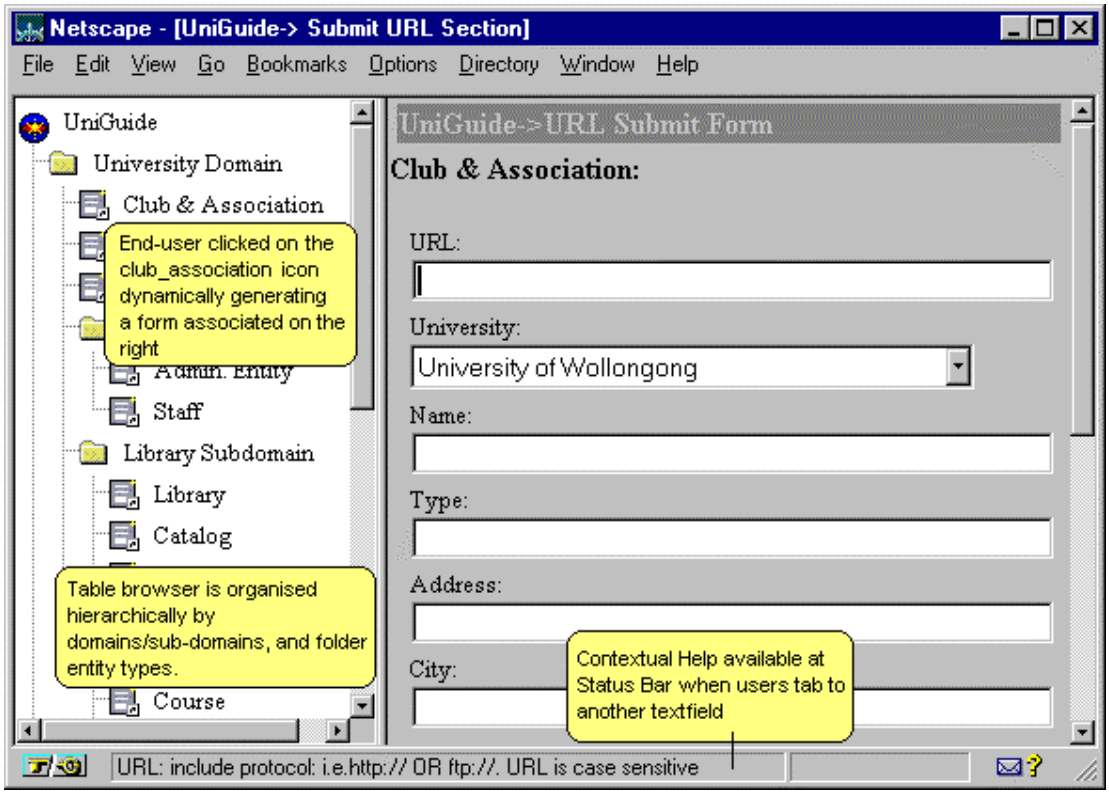

Figure 6: *UniGuide* Table/Set Browser: Submit URL Section.

4.2 Query Section

4.2.1 Simple Queries

A simple query form is shown in Figure 7. Entities are grouped hierarchically.

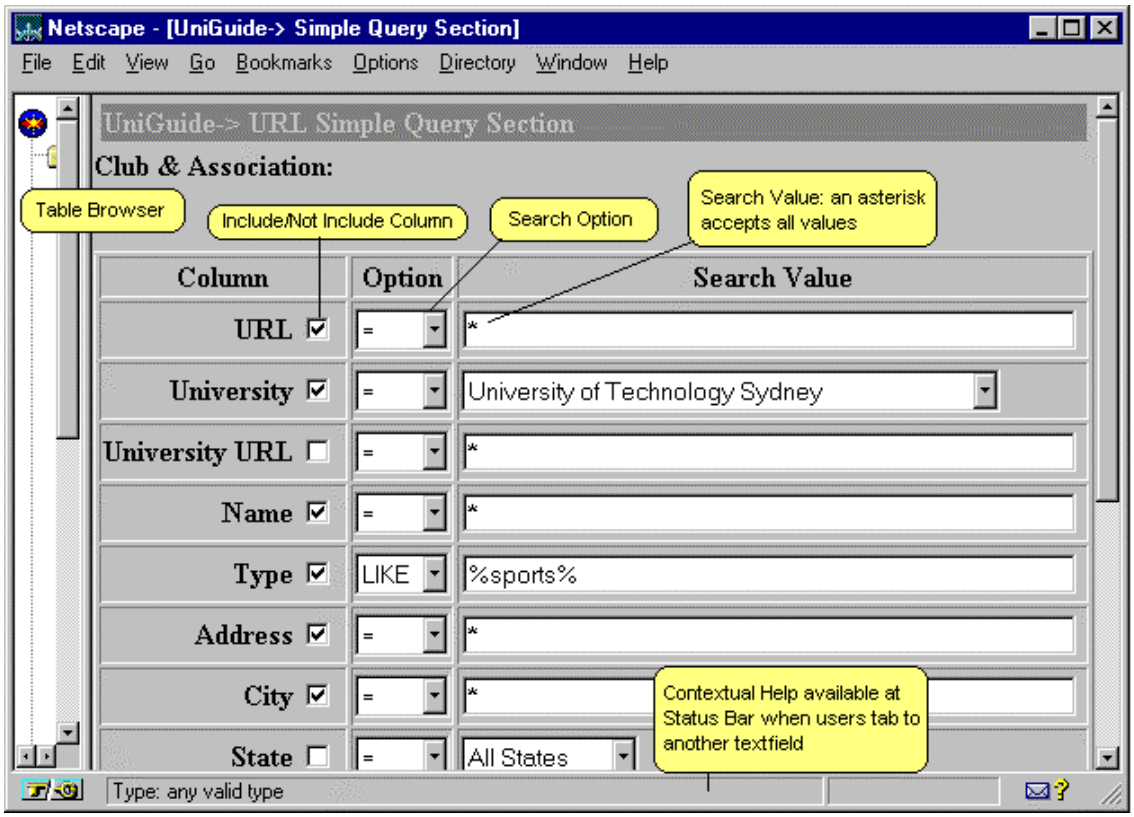

Figure 7: *UniGuide* Simple Query Form Interface. Note: (*) accepts all values.

When end-users click on a given entity, an HTML form is generated dynamically on the right hand-side. End-users can specify the range of values to search in the text boxes. The search options are contextual depending of the type of data in the column (i.e. LIKE option is activated only for columns of type text). The output of a query is displayed in a tabular form. This kind of user interface can provide even casual users a rapid overview of the schema to which she/he can get accustomed easily.

4.2.2 Configurable Queries: FreeSQL

The FreeSQL form is shown in Figure 8. End-users can configure and customise the required query. A more complex interface is provided for advanced end-users that allows the construction of more complex SQL queries with the aid of predefined queries, functions, operators, and a list of tables and columns available in the schema.

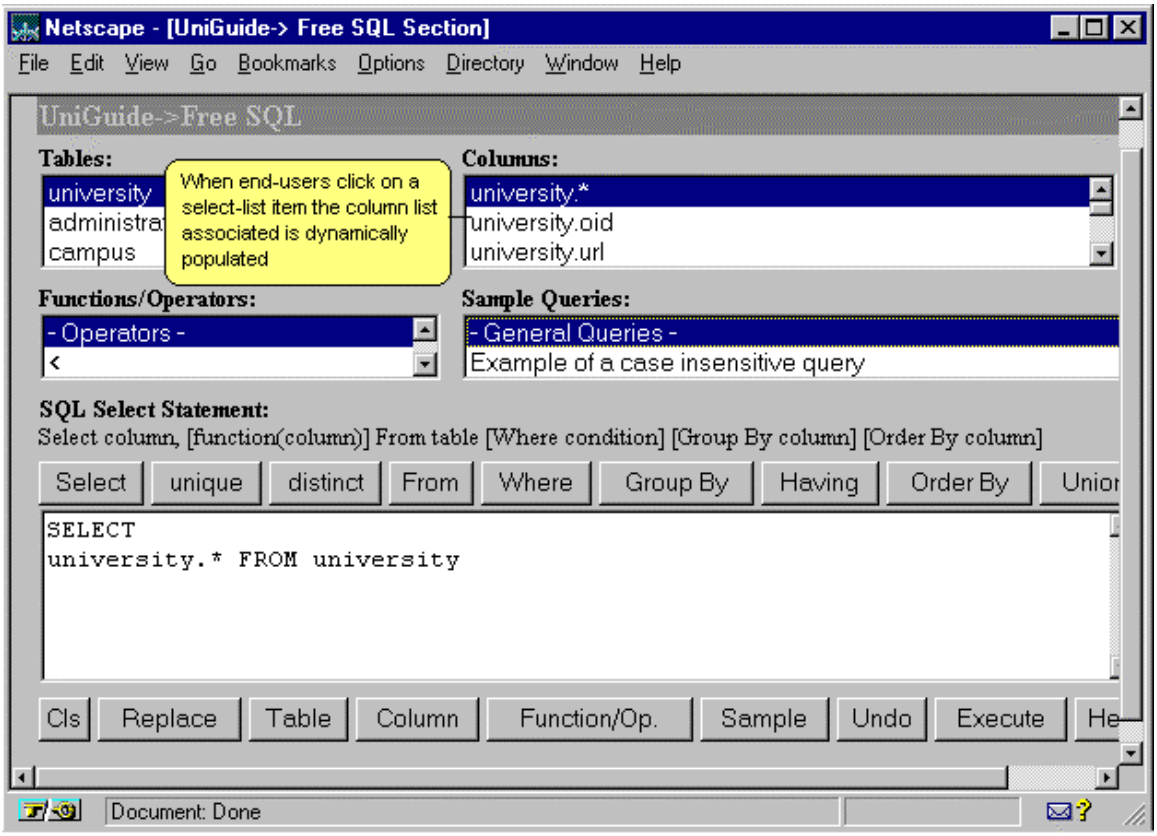

Figure 8: *UniGuide* FreeSQL interface

4.2.3 Query Taxonomy: examples of different types of queries

We distinguish between intra-site queries (queries which involve a particular university) and inter-site queries (queries that span multiple universities, eg. all universities in the state of New South Wales). Also, we can classify queries on a continuum from simple to complex. We have identified the following query types (see table 2): simple queries (one entity/set involved), summarised information (aggregate functions: average, sum, count, etc.), joins (more than one entity/set involved) and simulation of transitive closure (over unary relations). Examples of queries in each cell are presented below.

	Simple Query	Summarised Information	Joins	Transitive closure
Intra-site	Q1	Q3	Q5	Q7
Inter-site	Q2	Q4	Q6	N.A.
	Simple ←			→ Complex

Table 2. Query Taxonomy

Examples of simple queries:

- ***Intra-site***

  Q1: Find all the home pages of university residential colleges of a particular university that offer full board and are coed.

  Classes/tables: residential_college

- ***Inter-site***

  Q2: Find all the relevant information of Information Systems departments located in the state of NSW.

  Classes/tables: academic_entity

Examples summarised information:

- ***Intra-site***

  Q3: Find the number of research entities associated with a particular University

  Classes/tables: university, research_entity

- ***Inter-site***

  Q4: Find the number of publications printed in 1998 and related to databases by university for all the universities.

  Classes/tables: university, academic_entity, research_entity, _academic_entity_spublications, _research_entity_spublications

Examples joins:

- ***Intra-site***

  Q5: Find the e-mail addresses of all staff members in the Computer Science department of a particular university who are interested in Web-related research.

  Classes/tables: academic_entity, _academic_entity_sstaff

- ***Inter-site***

  Q6: Find the URLs of academic entities and research entities involved in a particular project funded by a particular agency and started in the current academic year.

  Classes/tables: academic_entity, research_entity, _academic_entity_sprojects, _research_entity_sprojects

Example simulation of transitive closure:

- ***Intra-site***

  Q7: Find and retrieve all the relevant information of all subordinate academic entities of the Faculty of Commerce of a particular university

  Classes/tables: academic_entity

- ***Inter-site***

  Not applicable.

5. Conclusion

A new kind of search engine has been proposed as an alternative to current implementations, with the ability to provide more structured and complex queries. The deployment of schema-based database search engines enables the definition of a new kind of resource discovery over Web data: the transformation of well-demarcated sub-Webs on the WWW into highly structured databases. This work is part of an ongoing research program exploring object-relational database approaches to searching the Web. The success of this project is predicated on the agreement by the target universities to adopt the use of *UniGuide* Scheme metadata in order to populate the *UniGuide* database automatically. We would like to emphasise that although the proposed solution is domain-specific, wherever a model can be "extracted" and a standard can be established for metadata, such as in corporate IntraNets and ExtraNets, our approach can be customised to adapt to the requirements of that specialised domain. Finally we conclude that due to the exponential growth of the WWW and the resulting complexity of the resource discovery problem, there is a future for more focused and specialised search engines which cater to coherent subsets of the World Wide Web.

References

[Abiteboul 1997]

Abiteboul, S. "*Querying Semi-Structured Data",* ICDT 97 6th International Conference on Database Theory Delphi, Greece, January 8-10, 1997.

ftp://ftp.inria.fr/INRIA/Projects/verso/VersoReport-103.ps.gz

[Ashish and Knoblock 1997]

Ashish, N., and Knoblock C. "*Wrapper generation for semi-structured internet sources",* Proceedings of the Workshop on Management of Semistructured Data In conjunction with PODS/SIGMOD-97, Ventana Canyon Resort, Tucson, Arizona 1997

[Atzeni et al. 1997]

Atzeni, P.; Mecca, G; Merialdo, P.; and Tabet, E. "*Structures in the Web",* Technical Report RT-INF-19-1997, Department of Computer Science and Automation, January 1997.

http://www.inf.uniroma3.it/tech-rep/inf-19-97.ps

[Berners-Lee et al. 1994]

Berners-Lee; T., Cailliau, R.; Luotonen, A.; and Nielsen, H. F., Secret, A. "*The World Wide Web"*, CACM, v37(8), August 1994, pages 76-82

[Bharat and Broder 1998]

Bharat, K., and Broder, A. "*A technique for measuring the relative size and overlap of public Web search engines",* Proceedings 7th International World Wide Web Conference (WWW7), Brisbane, April 14 1998

http://www7.scu.edu.au/programme/fullpapers/1937/com1937.htm

[Hammer et al. 1997]

Hammer, J.; Garcia-Molina, H.; Cho, J.; Aranha, R.; and A. Crespo, "*Extracting Semi-Structured Information from the Web*", Proceedings of the Workshop on Management of Semistructured Data, In conjunction with PODS/SIGMOD-97, Ventana Canyon Resort, Tucson, Arizona 1997

[Han et. al 1995]

Han, J.; Zaïane, O. R.; and Fu Y. "*Resource and Knowledge Discovery in Global Information Systems: A Scalable Multiple Layered Database Approach"*, Proceedings Of a Forum on Research and Technology Advances in Digital Libraries (ADL'95), McLean, Virginia, May 1995.

ftp://ftp.fas.sfu.ca/pub/cs/han/kdd/gnis94.ps

[Mahalingam and Huhns 1997]

Mahalingam, K, and Huhns, M. "*A tool for Organising Web Information"*, IEEE Computer, June 1997, pages 80-83.

[OCLC 1997]

OCLC Online Computer Library Center Inc. *Dublin Core Metadata Element Set: Reference Description*. November 1997.

http://purl.org/metadata/dublin_core_elements

[Poulter 1997]

Poulter, A. "*The design of World Wide Web search engines: a critical review",* Program, vol31 no.2 April 1997, pages 131-145.

[Rajamaran 1998]

Rajamaran, A. "*Transforming the Internet into a Database",* Workshop on Reuse of Web information hold in conjunction with the 7th International World Wide Web Conference (WWW7), Brisbane, April 14,1998

http://www.mel.dit.csiro.au/~vercous/REUSE/pos8/index.html

[Stonebraker and Moore 1996]

Stonebraker, M., and Moore, D. *Object-Relational DBMSs: The Next Great* Wave. Morgan Kaufmann Publishers, Inc 1996

[Sunderraman 1997]

Sunderraman, R. "*Relational Querying of Semi-structured data",* Proceedings of the workshop on Information Technology and Systems (WITS' 97), Atlanta December 13-14, 1997 (pg. 11-20).

[Vercoustre 1998]

Vercoustre, A. M. "*Reuse of Web Information",* Keynote address presented at the workshop on Reuse of Web information hold in conjunction with the 7th International World Wide Web Conference (WWW7), Brisbane, April 14,1998

[W3C 1998a]

W3C. "*HTML 4.0 Specification W3C Recommendation",* The global structure of an HTML document. Metadata, April 1998.

http://www.w3.org/TR/REC-html40/struct/global.html

[W3C 1998b]

W3C. "*Resource Description Framework (RDF) Homepage"*, April 1998.

http://www.w3.org/RDF/