A FRAMEWORK FOR THE ORGANIZATION AND DISCOVERY OF INFORMATION RESOURCES IN A WWW ENVIRONMENT USING ASSOCIATION, CLASSIFICATION AND DEDUCTION

by

Marut Buranarach

B.E., King Mongkut's Institute of Technology Ladkrabang, Thailand, 1994

M.S., University of Pittsburgh, 1999

Submitted to the Graduate Faculty of

Information Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2004

UNIVERSITY OF PITTSBURGH

INFORMATION SCIENCES

This dissertation was presented

by

Marut Buranarach

It was defended on

December 13, 2004

and approved by

Stephen C. Hirtle, PhD, Professor

Ronald L. Larsen, PhD, Professor

Janyce M. Wiebe, PhD, Associate Professor

Vladimir I. Zadorozhny, PhD, Assistant Professor

Dissertation Director: Michael B. Spring, PhD, Associate Professor

A FRAMEWORK FOR THE ORGANIZATION AND DISCOVERY OF INFORMATION RESOURCES IN A WWW ENVIRONMENT USING ASSOCIATION, CLASSIFICATION AND DEDUCTION

Marut Buranarach, PhD

University of Pittsburgh, 2004

The Semantic Web is envisioned as a next-generation WWW environment in which information is given well-defined meaning. Although the standards for the Semantic Web are being established, it is as yet unclear how the Semantic Web will allow information resources to be effectively organized and discovered in an automated fashion. This dissertation research explores the organization and discovery of resources for the Semantic Web. It assumes that resources on the Semantic Web will be retrieved based on metadata and ontologies that will provide an effective basis for automated deduction. An integrated deduction system based on the Resource Description Framework (RDF), the DARPA Agent Markup Language (DAML) and description logic (DL) was built. A case study was conducted to study the system effectiveness in retrieving resources in a large Web resource collection. The results showed that deduction has an overall positive impact on the retrieval of the collection over the defined queries. The greatest positive impact occurred when precision was perfect with no decrease in recall. The sensitivity analysis was conducted over properties of resources, subject categories, query expressions and relevance judgment in observing their relationships with the retrieval performance. The results highlight both the potentials and various issues in applying deduction over metadata and ontologies. Further investigation will be required for additional improvement. The factors that can contribute to degraded performance were identified and addressed. Some guidelines were developed based on the lessons learned from the case study for the development of Semantic Web data and systems.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACD	Association, Classification and Deduction
DL	Description Logic
DAML	DARPA Agent Markup Language
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
KR	Knowledge Representation
RDF	Resource Description Framework
RDFS	RDF Schema
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
XML	Extensible Markup Language

1.0 INTRODUCTION

1.1 OVERVIEW

The World Wide Web may be the largest and most accessible public knowledge base ever developed. It contains a large number of information resources covering almost every subject and accessible by anyone with an Internet connection. With the increasing number of information resources on the Web, it is often more difficult to locate the resources that are relevant to a given need. New mechanisms and tools to help people to find relevant resources on the Web are needed.

The Web is organized at a base level by hypertext links. Individuals can link one resource to other resources, either on the same or different machines. This kind of organization resembles the notion of organization by association envisioned by Bush [1]. Although links are simple and scalable from a networking point of view, they are not as effective in helping people to locate resources. As the number of links grows, there is no guarantee that similar resources will be proximally linked.

In contrast to organization by linking, classification and cataloging have been useful techniques in organizing and retrieving information resources in library. Classification allows library resources to be organized in a systematic order, i.e. by their subject areas. A library catalog provides essential facts about the library resources, e.g. identification, physical characteristics and subject headings, etc. This information allows users to locate and retrieve information resources from a library collection by subject, title, author, etc. The catalog serves as an intermediary and an alternative to searching by shelf location.

There have been attempts to organize the resources on the Web the same way library resources are organized. Metadata is a form of cataloging information for the electronic resources [2]. It provides descriptions of electronic resources for the purpose of classifying and retrieving them. The early efforts to utilize metadata for the resources on the Web include the HTML META tag [3], PICS [4] and the Dublin Core metadata element set [5] initiatives.

However, it is unlikely that a single cataloging and classification scheme, such as the Library of Congress Classification System or the Dewey Decimal System will be accepted and employed on the Web. For the Web, decentralized schemes, such as the Warwick Framework [6], have been proposed. They emphasize the need for multiple metadata standards to interoperate. The Resource Description Framework (RDF) [7] could be considered an implementation of such a framework. RDF provides the meta-language for the creation and utilization of metadata on the Web. RDF is considered as one approach to the deployment of the Semantic Web, which is a project of the World Wide Web Consortium[1] (W3C).

The Semantic Web is an extension of the World Wide Web, in which information is given well-defined meaning, better enabling computers and people to work in cooperation [8]. It provides a mechanism to augment the Web with metadata of web resources. However, unlike many previous metadata attempts which described resources using a single vocabulary, RDF envisions people creating their own vocabulary to describe their resources. It should be noted that RDF has been designed to facilitate computer program processing and is not intended for human consumption.

1.2 PROBLEM AND MOTIVATION

One of the major obstacles in finding the information on the Web is the decentralized and ad hoc organization of information resources. While the intentional nature of creating a link often results in clusters of similar resources, there is no guarantee that linked resources are semantically

[1] http://www.w3.org/

related. This has raised one of the fundamental questions: How can the Web be better organized for more effective retrieval of resources [9]? Although various techniques exist for the organization of printed materials, there have been difficulties in applying them to the resources in the distributed and volatile environment like the Web.

One goal of the Semantic Web research initiatives is to allow for a more effective discovery of information resources on the Web [10]. While the contributions from various fields of research have been significantly made to the evolution of the Semantic Web, including theoretical foundations, standards, tools and applications, to the best of our knowledge, none has established an integration framework for how the Semantic Web will provide a solution for the organization and discovery of information resources on the Web. This has been a motivation for this dissertation research in exploring such a theme.

1.3 OBJECTIVES AND SCOPE

The main objective of this dissertation is to explore the deployment of the Semantic Web in the context of its organization and discovery of information resources. The fundamental framework and its implications will be explored. The objective can be divided into four sub objectives as follows:

1) To examine some fundamental principles in the organization and discovery of information resources that the Semantic Web will be based on.
2) To review and describe some theoretical foundations of the Semantic Web and related research efforts.
3) To elaborate an implementation framework for the organization and discovery of information resources on the Semantic Web based on the foundations and standards of the Semantic Web established by W3C.
4) To develop a case study evaluating the effectiveness of the framework and provide some recommendations in deploying the framework using the results of the case study.

This research focuses on the assessment of the effectiveness of an integration framework. Some more research will be required before a complete framework can be established. The problem focused is on assessing impacts of the framework in a real-world setting. Issues related to semantics harmonization and user interfaces are not addressed in this dissertation.

1.4 ORGANIZATION OF THE PAPER

This dissertation is organized as follows. The remaining of this chapter describes the approaches in the organization and discovery of information resources using association, classification and deduction (ACD) and provides an introduction to the Semantic Web. The second chapter reviews and describes works considered a foundation for the Semantic Web research. The third chapter describes an integrated framework and system for processing information of resources on the Semantic Web. The fourth chapter describes a case study assessing the impacts of the system on the finding of resources in a large Web resource collection. The final chapter provides some guidelines based on the lessons learned from the case study for the development of Semantic Web data and systems.

1.5 DEFINITIONS

1.5.1 Information

Definition: There are various definitions of *Information*. They range from Shannon's measure of information at the number of bits required to communicate a message over a communications channel [11] to the common sense definition where *Information* refers to *facts*, e.g. what today date is, what my birth date is [12] (p.3). It is also possible to define information as a commodity exchanged between entities and to use economic measures of its value as suggested by Dertouzos [13]. Most importantly for this research is the fact that "information" regardless of how it is

measured, or precisely defined, may be captured in language, art, or imagery. Further, once captured in some durable form, that object represents an information resource.

1.5.2 Information resource

Definition: Information that resides in documents, information systems or other artifacts constitutes an information resource. Its meaning is fixed by its representation in the artifact [14] (p.3).

1.5.3 Knowledge

Definition: In this paper, the term *knowledge* refers to the following definition given by Debons et al.: "*Knowledge implies a state of understanding beyond awareness. It represents an intellectual capability to extrapolate beyond facts and draw original conclusions.*" [12] (p.3)

1.5.4 Classification

Definition: Classification is the process of grouping things or objects that have the property or characteristic in common into a class. In the context of information, classification is the act of organizing the universe of knowledge into some systematic order [15] (p.209).

1.5.5 Catalog

Definition: A catalog is an organized presentation of information resources in accord with one or more systems of classification. Thus, a catalog is the implementation of a classification scheme over some set of information resources. For example, the bibliographic records of a library collection represent a catalog. The information provided in the catalog allows a user to identify particular items in the collection or to select relevant items for specific purposes [15] (p.3).

1.5.6 Association

Definition: Generally, an association is a connection of persons, things, or ideas by some common factor [16]. In the context of this dissertation, the notion of association is based on Bush's notion of selection by association [1].

1.5.7 Deduction

Definition: Deduction is the deriving of a conclusion by reasoning [17]. This dissertation focuses on the form of automated deduction based on the declarative approach. The declarative approach attempts to formalize semantics and common-sense reasoning using formal logics [18]. Under such a framework, deduction could be viewed as an ability of a computer program to deduce new conclusions from the given facts.

This dissertation explores the uses of automated deduction to help in the organization of information resources. More specifically, deduction is used to examine the relationships of information resources and their grouping into classes. The main deduction task for such purpose is *subsumption*. The definition of *subsumption* is formally provided as: *class A subsumes class B if every object that is a member of class B is also a member of class A*.

Alternate approaches to the logic-based deduction include the connectionist and the probabilistic approaches. The connectionist model usually derives a conclusion based on the mathematical properties of the interconnected units. One of the major techniques used in the connectionist model is *backpropagation*. The probabilistic model derives a conclusion given the uncertainty of information. One of the major techniques used in the probabilistic model is *Bayesian network*. These forms of automated deduction are beyond the scope of this research.

1.5.8 The Semantic Web

Definition: The Semantic Web is an extension of the World Wide Web, in which information is given well-defined meaning, better enabling computers and people to work in cooperation [8].

The Semantic Web is the evolution of the World Wide Web in such a manner as to maintain the structure and accessibility that exists currently while adding new features that adhere to the architectural design principle of the Web effort. It allows for both classification of web resources in a more rigorous fashion and machines access and manipulation of web resources in a reliable fashion. The Semantic Web is likely to be based on the RDF standards and other standards to be defined. One approach to the Semantic Web is being developed by the W3C, in collaboration with a large number of researchers and industrial partners [10].

1.6 THE ORGANIZATION AND DISCOVERY OF INFORMATION RESOURCES USING ASSOCIATION, CLASSIFICATION AND DEDUCTION (ACD)

There are a number of approaches to organizing information. Historically, classification has been the dominant approach to the organization of information. The rise of the World Wide Web has introduced association or linking as an alternative method for decentralized and ad hoc organization of information. This dissertation elaborates a combination of these forms with deduction to provide a more comprehensive system of organization for collections.

1.6.1 Classification of Information

The history of classification began with the establishment of the first library at the port of Alexandria in 285 B.C. [19]. Ptolemy I (Ptolemaios Soter) was persuaded by Demetrios Phalereus to collect copies of all known books to the library of Alexandria. With a growing set of resources in the library, books and scrolls were kept in piles or pits in order to group like materials together. The first organization of the materials was modeled after Aristotle's divisions of knowledge: mathematics, medicine, astronomy and geometry. Classification has become the major approach in the organization of printed materials, i.e. library resources.

According to Chan [15], the process of classification begins with the universe of knowledge and divides it into successive subcategories. The progression starts from the general ones to the specific ones. A classification scheme usually forms a hierarchical structure. The classes within each stage are often mutually exclusive.

Although classification has been effective in the organization of information, assigning information resources into appropriate categories are often difficult. This becomes even more complicated when different people classify things differently. This results in the existence of various classification schemes. For example, the Dewey Decimal Classification (DDC) [20] and the Library of Congress Classification (LCC) [21] are among the most widely used classification schemes in library classification.

The complication of the organization and discovery of information using the classification approach could be illustrated using the bucket metaphor shown in the Figure 1.1.

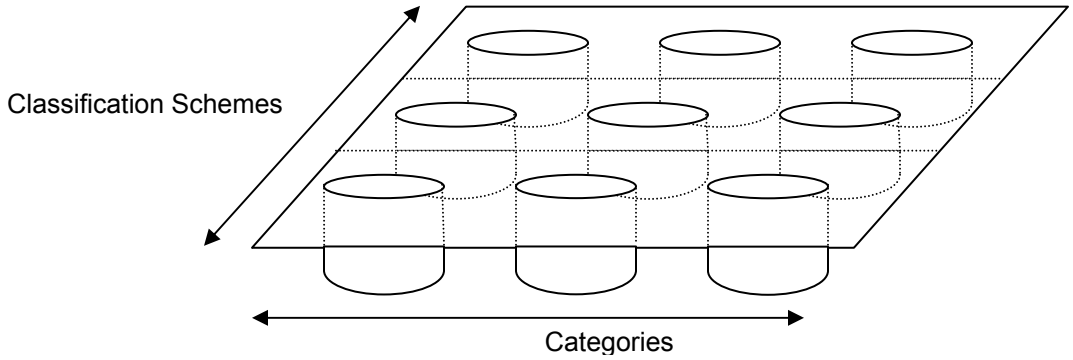


Figure 1.1. Problem of information classification

1.6.2 Association of Information

An information resource can also be organized in terms of its associations. In the famous "As we may think" [1], Bush emphasized that this form of organization conforms to how the human mind operates. He advocated organization by association where information resources are stored, discovered and retrieved by their relatedness.

Association of information could happen in many forms. In the context of this dissertation, associations may be undirected or directed, unidirectional or bi-directional and untyped or typed. For example, resources on the World Wide Web are organized by association. The association represents a hypertext system that is directed, where the direction of the link or association is defined from the source page to the destination page. It is unidirectional, where the link from the source page could lead to the destination page but not vise versa. The type of the link is generally undefined or untyped, which means the specific meaning of the relatedness between the source and the destination pages is not defined.

In scientific publications, citation is a commonly used reference mechanism. Citation represents a form of association between information resources that could lead to the discovery of relevant information. In particular, citations allow readers to discover related publications referenced in a publication. The form of the association established by the citation mechanism is similar to the one provided in the hypertext on the Web, i.e. directed, unidirectional and untyped[1].

Association that is directed, bi-directional and typed is the most expressive form of association. That is the association not only indicates the existence of the relatedness but also the specific information of the mutual relationships. For example, in some traditional library catalogs, record of book title may refer to record of book author while record of book author may inversely refer to record of book title. The association between the records of book title and author in such a catalog exemplifies a form of directed, bi-directional and typed association.

1.6.3 The Reality: Mixed Models of Classification and Association and Unresolved Problems

Classification and association could also augment each other in the organization and discovery of information resources. For example, the use of cross-references in library classification systems

[1] Although one might consider that there is a meaning underlying in the citation association, i.e. 'reference to', its type is considered undefined because no other meaning can be represented by the association.

represents a form of association that could help in the finding of relevant information in classification system. Similarly, classification system is often helpful in association system. For example, in a large association system such as the World Wide Web, classification systems, e.g. the Internet directories, are often needed.

The organization by association is sometimes classificatory in nature and the organization by classification is sometimes associative. Put in a less formal way, the related objects are often grouped in the same cluster and the objects in the same cluster are often related. Thus, classification systems may include associations and association systems may be focused on classifying resources.

Even when information is well classified and related, it may not be able to be discovered. Swanson refers to it as the problem of undiscovered public knowledge [22]. One of the examples illustrated by Swanson relates to the reports on the relationships between fish oil, blood viscosity reduction, and improvement of Raynaud's disease patient. The example was based on two medical reports: one provided a report that the use of fish oil could result in reducing blood viscosity while the other independently reported that blood viscosity reduction could result in some improvement of Raynaud's disease patient. Although the reports might suggest an implicit relationship between fish oil and improvement of Raynaud's disease patient, the conclusion may not be reached unless both reports were known. Swanson's example is a very primitive example of the existence of the implicit information, which could be undiscovered.

A similar problem in association system could be demonstrated using the example of citations. While a citation establishes reference relationship from the referring literature to the referred literature, it does not establish the relationship from the referred literature to the referring one. As a result, the discovery of the referred literature may not lead to the discovery of the referring literature even though they may be relevant. While citation searches allow this reverse relationship to be discovered, the process is often costly and imperfect.

1.6.4 Deduction of Information

Deduction refers to an ability of a computer program to derive conclusions from the given facts [18]. In the context of this dissertation, deduction can supplement classification and association in providing a more comprehensive system of organization for collections of information resource. In particular, deduction provides automated classification that can help in reducing the effort in classifying information resources. The automation provided by deduction will allow for a more versatile organization of information resources that can lead to better resource discovery.

Some relationships between information resources may not be explicitly stated but could be inferred based on the given information. For example, the problem illustrated in the citation example presents the limitation existing in many association systems. As an association between two information resources is created, another relationship in the inverse direction always exists implicitly. Automated deduction about resource relationships may be needed to ensure that implicit relationships between information resources will be discovered.

1.7 THE ORGANIZATION OF INFORMATION ON THE WORLD WIDE WEB

Fundamentally, information resources on the Web are organized by association, i.e. hypertext links. Although links are simple and scalable, the organizational structure is not necessarily effective in helping people to locate resources. As the number of links grows, it is easy to get lost and not able to find the needed information. Further, small world network research [23], suggests that link traversal rapidly expands the selected set beyond any manageable size. For example, following all links from a given page through ten stages with the average of five links per page results in an average retrieved set size at the magnitude of ten million pages (5^{10}).

Internet directories, such as Yahoo![1] and the Open Directory Project[2], provide a form of classification for web resources. The directories are usually created by human experts in the subject areas. This is to provide accuracy and consistency in resource organization. However, as the numbers of resources on the Web grows, it is difficult to maintain the directories. A large number of resources are overlooked by the directories. As a result, many useful resources are not included in directories.

Search engines, such as Google [3] and AltaVista [4], provide an automated mechanism in indexing web resources. A search engine organizes resources by the text they contain. It stores an index of words and pointers to the resources that contains these words. The index is generated by the crawler which routinely gathers the information from a pre-defined list of resources. Internet users can run a query against the search engine that will return the list of resources whose keywords match with the query. Although the coverage of resources covered by search engines is far greater than those of the directories, the results are generally not guaranteed to be relevant to the user queries. Furthermore, the full-text indexing strategy offered by the search engine is not applicable to non-text resources, such as images or executable programs.

1.8 THE SEMANTIC WEB

The Semantic Web is an extension of the Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation [8]. From the architectural viewpoint, the Semantic Web is the evolution of the World Wide Web in such a manner as to maintain the structure and accessibility that exists currently, while adding new features that adhere to the architectural design principle of the Web effort. The W3C Semantic Web Activity Statement [10] also includes the following explanation of the Semantic Web:

---

[1] http://www.yahoo.com/

[2] http://www.dmoz.org/

[3] http://www.google.com/

[4] http://www.altavista.com/

The Semantic Web is a vision: the idea of having data on the Web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications.

The term *Semantic Web* was coined by Tim Berners-Lee, the director of the W3C, who described it simply as "*a Web of data that can be processed directly or indirectly by machines"* [24]. The architecture of the Semantic Web as envisioned by Berners-Lee is shown in Figure 1.2 [25]. The architecture could be briefly introduced as follows.

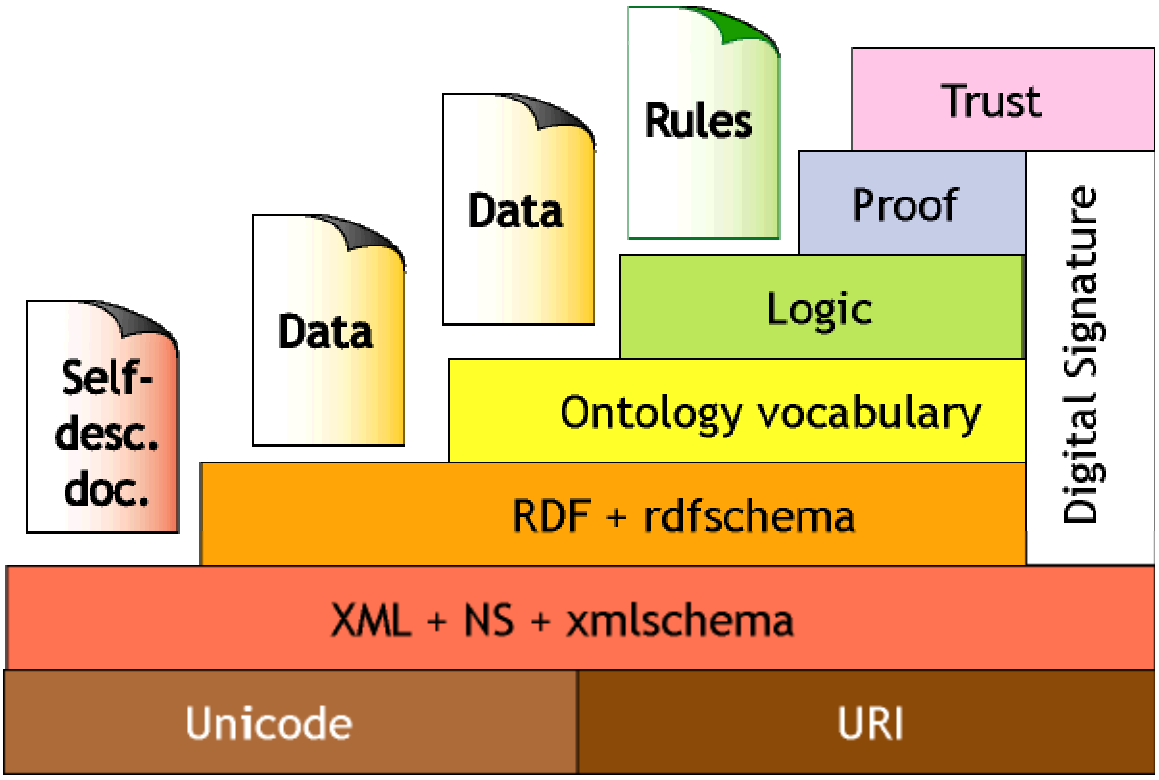


Figure 1.2. The Semantic Web architecture by Berners-Lee

At the base levels, the Extensible Markup Language (XML) [26] and the Resource Description Framework (RDF) [7] are considered to play major roles to the Semantic Web effort [27]. RDF utilizes XML as its composition layer.

Although XML allows different schema to be specified, which allows document instances to be created, different XML documents using different XML schema could be used to represent

the same meaning. This makes it difficult for the computer program to interpret the semantic equivalence of the two XML documents. RDF provides a mechanism to define the descriptions of Web documents in the form of metadata. RDF defines a uniform data model in the form of RDF triple: *Predicate*, *Subject* and *Object*. This canonical form makes it possible for the computer program to interpret the descriptions of Web documents independently of the syntax. The definition and relationship of terms used in an RDF statement are defined in an RDF Schema. The RDF Schema specification [28] specifies how RDF Schema could be created and processed.

To allow the data to be manipulated by computer programs, ontology and logic layers are required on top of the RDF layer. The ontology layer is needed to provide common terminologies in a domain of knowledge. The logic layer provides formal semantics, which is important to the machine-automated process such as making inferences.

Proof and the Web of trust, using digital signatures, are the mechanisms proposed to prevent inconsistency on the Semantic Web [29]. The investigation of these two layers is beyond the scope of this dissertation.

1.9 THE WEB AND THE SEMANTIC WEB

The Semantic Web and the World Wide Web share some common properties. Both rely on the Hypertext transfer protocol (HTTP), Uniform Resource Identifier (URI) and markup languages. While the Web was designed for human consumption, the Semantic Web provides an additional layer of information designed for machine processing. Full-text documents in HTML format are the main content for the Web. Metadata, in the form of resource descriptions and ontologies supplement this content in the Semantic Web. Search engines were created to help find information on the Web. New systems operating on metadata will be created to help find information on the Semantic Web. Although search engines operating on full-text documents and those operating on metadata are both designed to help the users to locate information, they will operate on different layers of information.

1.9.1 HTML Documents, Information Resources, and Web Resources

Web pages, i.e. HTML documents, are the most common form of information resources on the Web, but new forms based on programs are growing in proportion. The Semantic Web will provide metadata for information resources regardless of their form. In this way, the Semantic Web will allow for the description of "traditional" Web resources – html pages, dynamic resources – scripts and programs, and all other information resources, both digital and analog, -- books, articles, recordings, etc.

According to the RDF specification, a *Web resource* is an object that has a resource identifier in URI syntax. Within such a definition, a *Web resource* is not necessarily an *information resource*. For example, a person could be identified by an URI identifier and would be defined by RDF as a *Web resource*. Web resources that contain the information about the person, such as resume, personal homepage, images, etc., would be considered information resources on the person. Although the focus of the discussion is not on how one should differentiate non-information resources from information resources, it emphasizes that Web resources as defined by RDF are a broader category than information resources.

1.9.2 Full-text Indexing vs. Subject-based Classification

The Web contains many documents in HTML format. The search for information on the Web often relies on search engines that use full-text indexing techniques. This results in a simple classification of Web documents based on words. Such classification techniques have shortcomings. In particular, a single word could convey several meanings. For example, indexing documents with the keyword "Java" may include documents related to the programming language Java as well as the documents related to the Java islands of Indonesia. There is no guarantee that the documents returned are semantically related, full text indexing is not the most effective basis for accurate information retrieval.

The Semantic Web will use something close to subject-based classification. Subject-based classification provides the grouping of resources based on resource calegories. In particular,

resources will be grouped into categories based on some criteria beyond the words in the documents. For example, with subject-based classification, ideally the documents on Netherlands will be grouped into the same category whether the documents contain the terms “Netherlands”, “Holland”, “Dutch” or none of these terms. With the assumption that resources in the same category will be semantically related, subject-based classification provides for more accurate information retrieval.

Subject-based classification is not new for the Web. Yahoo! and other similar Internet directory services provide subject-based classification. Subject-based classification systems on the Web are usually created by some central authorities. The Semantic Web aims at the classification of Web resources that will occur in decentralized fashion. In particular, under the Semantic Web framework, classification of resources will occur as resources are posted. For such a classification to become pervasive, a more automated form of classification will be required.

1.9.3 Hypertext Links vs. Semantic Associations

The Web uses hypertext links to define associations between Web documents. Such associations can be organized or unorganized. For example, one can create hypertext links from a Web page to other Web pages having related subjects – perhaps the most common example is a single paper that is broken up into a series of linked sections. In such a case, the links facilitate the discovery of information using association. However, hypertext links can also be used to create random paths between Web documents. Such forms of association will not necessarily contribute to the discovery of information.

The Semantic Web will define links between Web resources based on semantic relationships. In particular, RDF data will establish virtual links for Web resources. Such links establish semantic associations between Web resources. With the linked resources related by definition, the effective discovery of resources based on associations could be achieved.

1.10 CONCLUSION

This chapter has discussed the motivation of the Semantic Web in the context of the organization and discovery of information resources using ACD. The next chapter will review and describe some theoretical foundations for the Semantic Web.

2.0 BACKGROUND

2.1. INTEROPERABILITY IN INFORMATION SYSTEMS

This section provides background on interoperability in information systems. Definitions and approaches for achieving interoperability in information systems are provided as follows.

2.1.1 Definition

Interoperability, as defined by the Institute of Electrical and Electronics Engineers (IEEE) [30], is "the ability of two or more systems or components to exchange information and to use the information that has been exchanged". A system (or component) can interoperate with any other system (or component) as long as they can understand the information that has been exchanged between each other.

Brodie [31] gives a functional definition of *Interoperability* in information systems as follows:

"Interoperability: Two components (or objects) X and Y can interoperate (are interoperable) if X can send requests for services (or messages) R to Y based on a mutual understanding of R by X and Y, and Y can return responses S to X based on a mutual understanding of S as (respectively) responses to R by X and Y." (p.13)

In short, system X can interoperate with system Y if X's requests can be responded to appropriately by Y.

Under the definitions from Brodie and IEEE, interoperability between information systems is not determined by the physical location. Thus, interoperability between two information systems (or components) can occur within the same machine or between two different machines. It is also not determined by the purpose of interoperation. Interoperability is only determined by the success of information exchange between two information systems.

2.1.2 Approach to Achieve Interoperability

2.1.2.1 Standards Interoperability is difficult when the number of different systems involved is high. The number of data conversions necessary for a system to communicate with *n* systems is equal to *n x (n-1)*. However, if there is an agreed-on set of rules, aka standards, for information exchange, the number of data conversions necessary is reduced to 2*n* or *O(n)*. Thus, standards are critical to the success of interoperability of heterogeneous systems.

Vckovski has defined the requirements for a good standard as follows: expressivity, unambiguity, extensibility and acceptance [32].

2.1.2.2 Layered approach Agreement is the goal of every standardization process. However, the standardization process is usually costly and time-consuming. This has led to the layered approach, where standards are layered into multiple levels. Spring [33] suggested that, by making the standards modular by layering, the impact of changes in one standard can be isolated from others. The layered approach has been widely used in telecommunications for internetworking, i.e. OSI reference model, TCP/IP reference model.

Unlike telecommunications, currently there is no well-defined separation between data modeling layers [33;34]. This lack of clear separation has led to the redundant features on different layers. Melnik and Decker [34] have proposed the Information Model Interoperability (IMI) reference model to identify separation between data modeling layers. In the IMI model, data modeling can be divided into three layers: *Syntax*, *Objects* and *Semantics*.

2.1.2.3 Mediators Mediator architecture, first introduced by Wiederhold [35], is another approach toward interoperation of heterogeneous information systems. Mediator architectures are also known as "information brokers", "knowbots" and "software agents" [36]. The mediator is a layer that sits between the user application layer and the data source layer (Figure 2.1). The role of the mediator is to perform necessary transformation and data mapping between different data sources.

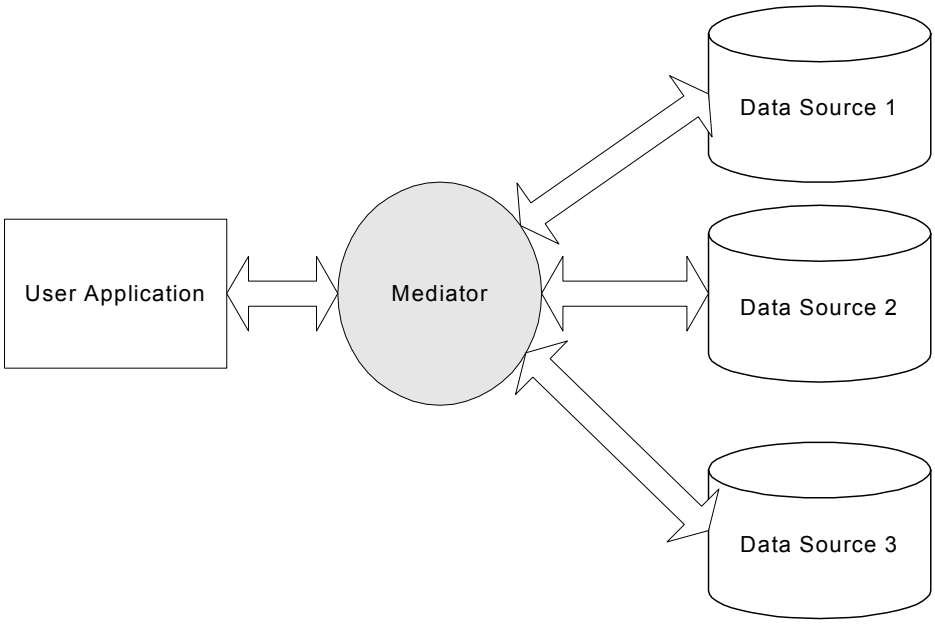


Figure 2.1. Mediator architecture

Mediator hides the heterogeneity of different data sources from the user applications. Thus it allows user applications to be independent of data sources. In order to perform data mapping, the mediator needs to have knowledge of user applications and data sources. For example, a mediator needs to understand the query formats used by user applications and data sources in order to map user's queries to the query formats that are required by data sources.

2.1.2.4 Wrappers When the number of data sources involved is high, the mediators knowledge about the data format of each data source can become unmanageable. In order to simplify the task of mediation, wrappers can be placed between the mediator and the data sources (Figure 2.2) by placing a wrapper around each data source. With wrappers, data sources will become homogeneous to the mediator. An example wrapper architecture can be found in [37]. Melnik

[38] describes the use of a canonical wrapper in combination with a mediator and layered architecture to facilitate the integration of heterogeneous information systems.

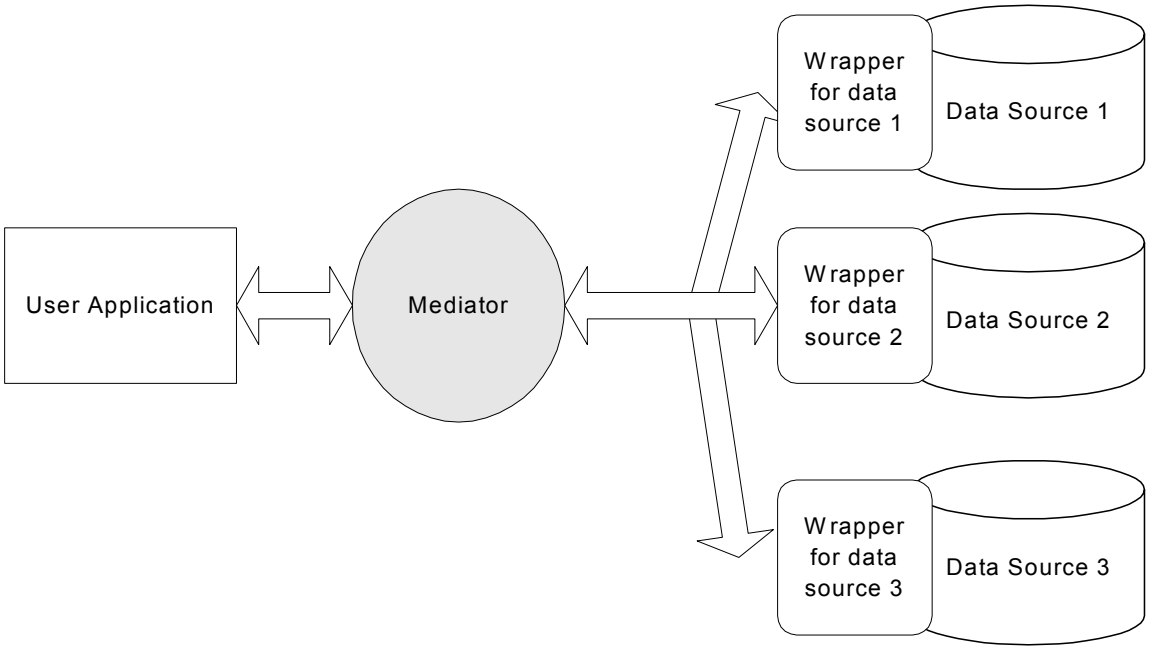


Figure 2.2. Wrapper architecture

2.2 KNOWLEDGE REPRESENTATION

This section provides background on Knowledge Representation (KR) research. Definitions of KR are given in section 2.2.1. Reviews of major approaches in KR are provided in sections 2.2.2-2.2.7

2.2.1 Definitions

Davis et al. [39] defined Knowledge Representation (KR) in terms of its five basic roles. The five basic roles of KR are fundamental properties that exist in all the invented representations. They are:

1) **A KR is a surrogate**. KR is a stand-in for things that exist in the real world. It is an attempt to find representations or surrogates of things. Thus KR provides surrogates, accepting that surrogates are, by definition, imperfect.
2) **A KR is a set of ontological commitments**. Ontological commitments are a set of decisions about what and how to represent the world.
3) **A KR is a fragmentary theory of intelligent reasoning**. In specifying a representation, it is also necessary to specify how to reason intelligently from it.
4) **A KR is a medium for efficient computation.** In order for machines to use a representation, i.e. reason about it, it must be able to make computations.
5) **A KR is a medium of human expression.** A representation is a language that humans can use to talk to the machine about the world. It is a medium of expression and communication from human to the machine.

2.2.2 Declarative vs. Procedural Knowledge

There have been two main approaches in representing knowledge for machine consumption: procedural (or imperative) and declarative knowledge. With the procedural approach, the machine is given instructions or procedures. These instructions are usually low-level. The outcome from the program is obtained by instructing the program to execute these instructions or procedures. This approach focuses on building semantics or knowledge of how to obtain the outcome. The declarative approach gives meaning by providing the program with facts it knows. This allows the machine to understand the meaning of a new thing by relating it to previous knowledge.

Even though the procedural approach of modeling semantics or knowledge is generally computationally faster, the declarative approach has several advantages, as suggested by McCarthy [18]. Using the declarative approach, the program can take advantage of previous knowledge. The program can determine the logical consequences of what it is told from what it previously knew. The meaning of declaratives is also less dependent on their order, thus making it easier to modify, i.e. after-thoughts. The declarative approach is also the form that is frequently used in human exchange; where the procedural form is currently used primarily to instruct

machines. McCarthy regarded the ability of the program to deduce immediate consequences of anything it is told, from what it already knows, as similar to human *common sense* [18]. The declarative semantics framework attempts to formalize semantics and reasoning using logics.

2.2.3 Intension vs. Extension

Philosophy distinguishes the expression of meaning into two categories: "*intension*" and "*extension*". The distinction is sometimes referred as "*sense*" and "*reference*" [40]. From a philosophical point of view, the *extension* is the set of all objects in the "actual" world that fall under the concept, whereas the *intension* is the set of objects that fall under the concept in "all possible worlds." In other words, intension could be considered as abstract meaning while extension could be considered as every individual that falls under the abstract meaning. For example, the intension of "human" is the characteristics that make an entity "human", while the extension of "human" consists of every person in this world. From a computer programming viewpoint, the distinction between intension and extension has been adopted in object-oriented programming model, where *Class* is analogous to "intension" while the set of *Objects* instantiated from the class is analogous to "extension".

Sowa [41] considers *Logic*, *Ontology* and *Computation* as three major components of KR. Logic provides the formalized language for the representation. Ontology provides the meaning and taxonomy of terms in the domain of interest. Computation is the implementation and manipulation for the computer.

2.2.4 Logic and Computation

Logic was first introduced by Aristotle (384-322 B.C.) in the form of syllogism as a simple form of inference. Leibniz (1646-1716) proposed an idea of using Mathematics to formalize logic. In 1879, Frege introduced *quantifiers* to allow the concise expression of facts about objects without enumerating them. First-order logic (FOL) or first-order predicate calculus (FOPC) has been the

most fundamental foundation for formalisms based on logic. Because of its expressiveness, logic has been widely used as a formalized language to represent knowledge.

FOL has been known as one of the most expressive and well-understood knowledge representation languages. A FOL *sentence* represents a fact, while an FOL *term* represents an object. It provides *logical operators* (*connectives*) for forming complex sentences. It also provides *quantifiers* to allow for the concise expression of facts about objects without enumerating them. Facts about objects are expressed in terms of *predicates*. An object can be referred to in term of its relation to other objects using *functions*. Symbols in FOL can be *variable* or *constant*. The syntax rules of FOL are shown in Table 2.1 [42]. Genesereth & Nilsson [43] and Russell & Norvig [42] provide good introduction and explanation of the syntax and semantics of FOL.

Table 2.1: Syntax rules of first-order logic in Backus-Naur Form (BNF)

Sentence →	*AtomicSentence* | *Sentence Connective Sentence* | *Quantifier Variable,…Sentence* | ¬*Sentence* | *(Sentence)*
AtomicSentence →	*Predicate(Term,…)* | *Term = Term*
Term →	*Function(Term,…)* | *Constant* | *Variable*
Connective →	⇒ | ∧ | ∨ | ⇔
Quantifier →	∀ | ∃
Constant →	*A* | X_1 | *John* | …
Variable →	*A* | *x* | *s*| …
Predicate →	*Before* | *HasColor* | *Raining* | …
Function →	*Mother* | *LeftLegOf* | …

For example, from the following two FOL sentences:

(1) On (BookOf(John), BookOf(Mary)) ∨ On (BookOf(Mary), BookOf(John))

(2) ∀x ∀y On(x,y) ⇒ Above(x,y)

The first sentence states that either the book of John is on book of Mary or the book of Mary is on book of John, where *On* is a predicate, *BookOf* is a function, *John* and *Mary* are constants and $\vee$ is the disjunction connective. The second sentence states that one thing that is on another thing implies one is above the other, where $\forall$ is universal quantifier, *x* and *y* are variables, *On* and *Above* are predicates.

Logic provides an ability to deduce new logical sentences from existing sentences using logical computation. This capability is referred to as logical inference or logical reasoning. A generalized pattern of inference is known as *inference rule* or *inference procedure*. Given an inference rule, one can derive a *conclusion* if the *condition* is met. Modus Ponens (MP) is an example of inference rule. Using Modus ponens, if A implies B (A $\Rightarrow$ B) and A is known to be true, one could draw the conclusion that B is true. From the second sentence in the previous example, if '*On(BookOf(John),BookOf(Mary))'* is known to be true, one can infer '*Above(BookOf(John), BookOf(Mary))'*. Other inference rules include Modus Tolens (MT), And Elimination (AE), And Introduction (AI), Universal Instantiation (UI), Existential Instantiation (EI), etc. [see [43] for details].

Evaluation of an inference procedure is given in term of its *soundness* and its *completeness.* An inference procedure is *sound* if and only if every sentence derived from the inference procedure is logically implied from the knowledge base. An inference procedure is *complete* if and only if all the sentences that could possibly be implied can be derived from the inference procedure. This means that the *complete* inference procedure is not only able to generate new logical sentences that make sense, but it must also be able to discover every logical sentence that could possibly be implied. The completeness of inference procedure is harder to achieve than soundness. Although all of the inference rules mentioned in the previous paragraph are *sound*, none of them are *complete*. Gödel, in his completeness theorem (1930-1931), showed that there exists a *complete* inference procedure in FOL. However, the procedure itself was not discovered until 1965 when the *resolution* algorithm was introduced [44]. The resolution algorithm proves a statement by showing that the negation of the statement produces a contradiction with the

statements that are known to be true in the knowledge base [see [43] and [42] for the explanation of resolution algorithm and examples].

For a given KR system, the task of determining whether a statement can be inferred from the given statements in the knowledge base can be computationally trivial or completely unsolvable. This computational ability is different from one knowledge representation language to another. This difference basically depends on the level of *expressiveness* of the language. It is harder to reason efficiently with a representation language when the degree of expressiveness of the language is high. This is known as a fundamental tradeoff between expressiveness and computational tractability of a knowledge representation language [45]. It is one of the most fundamental issues in the design and evaluation of a knowledge representation language.

Due to the high degree of expressiveness in FOL, reasoning in FOL is known to be an undecidable and intractable problem. Levesque & Brachman claimed that reducing the expressiveness of FOL to frame description form could lead to a better computability [45]. As a result, Brachman & Levesque [46] introduced a logic which aims to achieve computability by limiting the form of expression of FOL to frame description form. This has become a startup for a new branch of FOL that focuses on describing things and reasoning by determining the *subsumption* relationship. The logic is known as *Description Logic* (see section 2.2.7).

2.2.5 Ontology

In philosophy, ontology is the study of the nature and relations of being [17]. The term is used by KR community as a way of describing and representing things. The term is often associated with *formal ontology* or *mathematical ontology*, which is a symbolic description of things in a domain. Ontology in KR is a somewhat vague term, with different definitions given by various researchers. One of the most referenced definitions of Ontology in KR is “an explicit specification of a conceptualization” [47]. Gruber considers ontology as a representation of the knowledge of a domain, where a set of objects and their relationships are described by a representational vocabulary. Neches et al. [48] shares a similar viewpoint that ontology defines basic terms and relations using a vocabulary of a topic area as well as rules for extending the

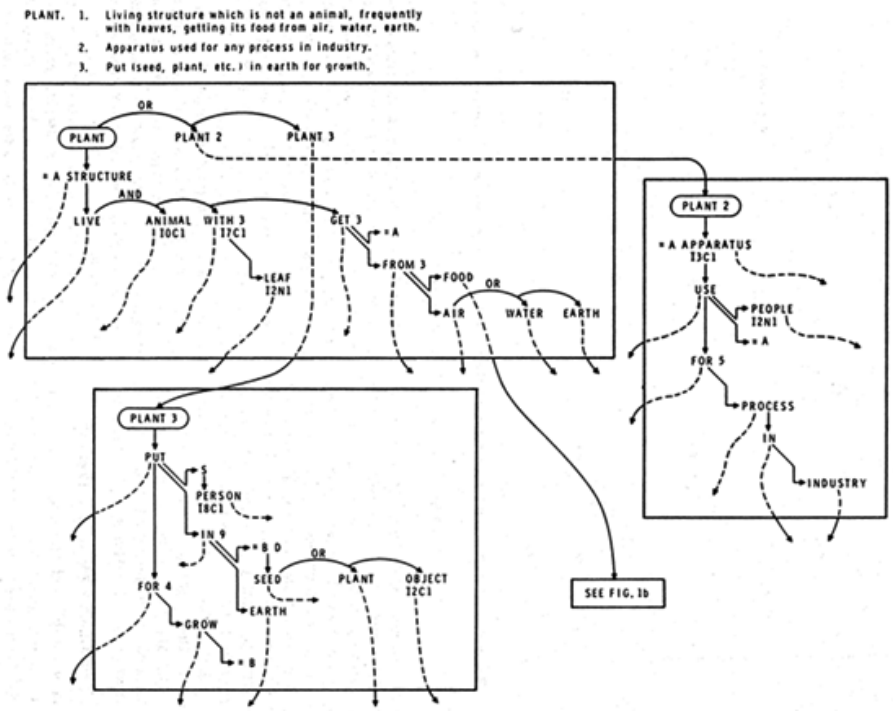
vocabulary. Ontology is used in KR to facilitate knowledge sharing and reuse. The effectiveness of the sharing and reuse depends on *ontological commitment*, which is an agreement to use the vocabulary in a coherent and consistent manner. Ontology is often related to taxonomy. According to Swartout et al. [49], Ontology is "a hierarchically structured set of terms for describing a domain that can be used as a skeletal foundation for a knowledge base". Sowa [41] provides a history and review of ontology work in KR.

2.2.6 Structured Approaches in Representing Knowledge

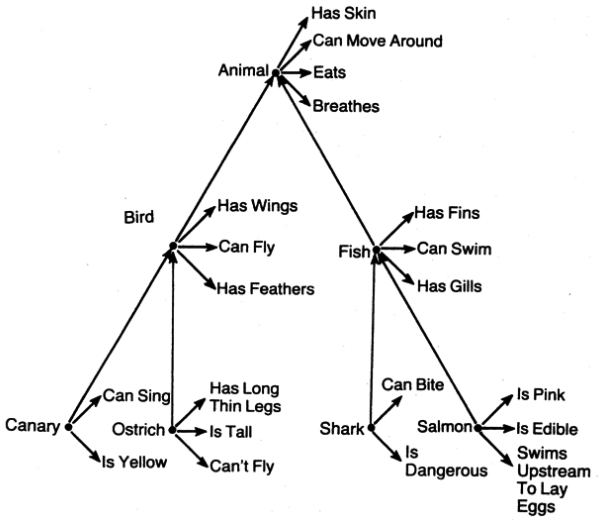
2.2.6.1 Frame-based systems One of the most influential knowledge representation schemes is *frames*. The notion of frames was first introduced by Minsky [50]. Frames were among the first *structured* knowledge representation approaches. A frame consists of slots (or attributes). Attached to each slot can be descriptions or procedures. The value attached to each slot can be filled by default or can have a value restriction. Collections of frames are organized and interconnected in frame systems.

Frames have been criticized for being too flexible and for lack of formalism. The formalism of frames has later been defined using *Description Logic* (see section 2.2.7)

2.2.6.2 Semantic networks The semantic network, or semantic net, was first introduced in 1966 by Quillian [51]. Semantic networks were used to represent word concepts in human memory. Semantic nets use a structured representation approach similar to frames. Semantic network can not only represent facts, but also associations between them. In essence, semantic networks contains links betwen facts. Some examples of semantic networks are shown in Figure 2.3.



a) a semantic network representing three meanings of "Plant" [51]



b) a semantic network representing a part of a simple animal hierarchy [52]

Figure 2.3: Examples of semantic networks

In 1975, Woods [53] argued that, "there is no "theory" of semantic networks". Woods suggested that people look at nodes and links in the semantic network without determining whether the meanings upon them are "abstract" (intension) level or "instance" (extension) level.

The distinction between intension and extension would help in avoiding unnecessary disagreement on the semantics of the network. Woods also made a distinction between structural links and assertional links, which depend on the sense of meaning being represented (intension or extension). Woods pointed out some limitations of semantic networks, including representation of relative clauses and quantification information. The "relative clause" problem relates to how to represent a reference to an entity that has already been referred to in the network. The quantification issue deals with the need to express quantifier information. The original semantic network is not expressive enough to represent these kinds of information.

Brachman [54] attempted to clarify the problem of not having uniform semantics for semantic networks, the issue that had been raised by Woods. Brachman suggested that the problem arises because various research efforts in semantic networks were based on different levels of primitives. According to Brachman, there were four different levels of semantic network primitives that the traditional semantic net research efforts were based on. These levels can be described as follows:

Implementational: Some programming-oriented research work views semantic nets at the level of processing units, i.e. data structures, pointers.

Logical: Most foundation research work in semantic nets treats logic and predicates (predicate calculus) as primitives.

Conceptual: Some research work deals with semantic nets at conceptual level, i.e. semantics and concept of words. This level is independent of language, taking the "thought influences language" viewpoint.

Linguistic: This view, in contrast to the conceptual view, uses the semantic net that is language-specific.

Brachman suggested the addition of a new level, "*Epistemological*", lying between the logical and conceptual levels. This level gives more a conceptual view to the logical level by adding the notions of inheritance, classification, etc. It also gives more formalism to the conceptual level by adding more structure to it.

Brachman designed a language, named KL-ONE, to reflect the idea of the epistemological level. KL-ONE proposes a formalism that operates at the epistemological level. One of the design goals of the KL-ONE language was to make the semantics of the language clear and well understood.

2.2.6.3 KL-ONE KL-ONE was originally known as the Structured Inheritance Networks (SI-Nets). It was first introduced in 1977 in Brachman's Ph.D. dissertation. KL-ONE represented knowledge at the "epistemological level", where the network is well structured and the type of each node is clearly defined. KL-ONE is not only a knowledge representation language; it also provides a utility for creation and query of the knowledge base. Thus KL-ONE can be considered as a knowledge representation "system". Brachman et al. [55] provides an overview of the KL-ONE system and its underlying framework. The description of KL-ONE that follows is based on it.

KL-ONE separates *assertion* expression from *description* expression. The *assertion* expression is a mechanism to make statement about things, such as the statement "I have a pen". The *description* expression deals with the description or definition of "object", such as the meaning of "pen". Assertion about things can be independent of the description of things. For example, asserting, "I drop a pen" or "I use a pen" does not change the description of "pen". The clear separation between assertional and descriptional component is one of the unique features of the KL-ONE language. The KL-ONE language focuses on the expression of *description.*

KL-ONE is an "object-centered" language, where "object" in KL-ONE can be one of the following three types: *Concept*, *Role* and *Individual*.

Concept. In KL-ONE, a concept can be *primitive* or *defined*. A concept is a *defined* concept, if it can be described necessarily and sufficiently in term of previously known concepts, otherwise it is a *primitive* concept. For example, if "bicycle" can be necessarily and sufficiently described in terms of the concept "vehicle" with 2 wheels, i.e. bicycle $\Leftrightarrow$ 2 wheel vehicle, then "bicycle" is a defined concept. In contrast, the concept of "taxi" can be necessarily described in

terms of the concept "vehicle" with 4 wheels, i.e. Taxi ⇒ 4 wheel vehicle, but not vice versa (4 wheel vehicle is not necessarily a taxi). In this case, "taxi" is a *primitive* concept.

Role. A role is property of concept. Role can be represented as attribute-value pairs. The value of each role attribute can be expressed in terms of relationships to known concepts and individuals. The constraints of these values can be given in terms of *Value Restrictions* (V/R). For example, the number of wheels of vehicle must be value-restricted to a number. The role's value restriction is similar to data type constraint of variable in programming languages.

Individual. An individual or an individual concept is similar to concept but can only be used to describe at most one individual. Individual can be considered as representation of instance of concept. For example, "John's bicycle", which refers to a specific instance of the "bicycle" concept, is an individual concept.

Figure 2.4 shows an example of the primitive concept of an e-mail message (MESSAGE) in KL-ONE. The diagram reads "A MESSAGE is, among other things, a THING with at least one Sender, all of which are PERSONs, at least one Recipient, all of which are PERSONs, a Body, which is a TEXT, a SendDate, which is a DATE, and a ReceivedDate, which is a DATE." [55]

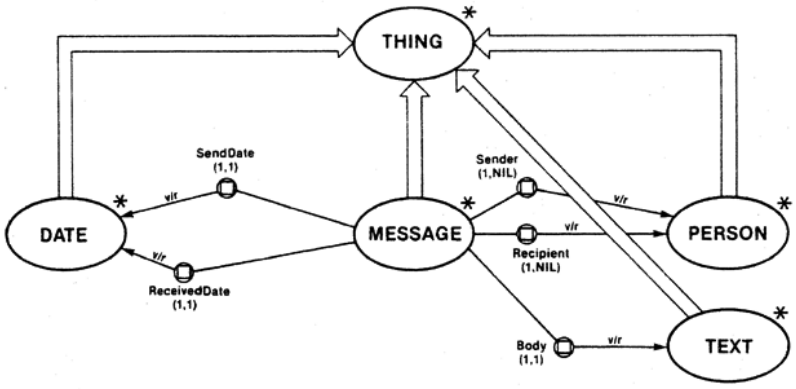


Figure 2.4 Example of a KL-ONE concept

A KL-ONE knowledge base can be considered as a type of semantic network with hierarchical organization representing inheritance relationships between concepts. The hierarchical organization is performed by determining *subsumption* relationships between

concepts. From the definition, Concept A subsumes Concept B if every individual of Concept B can be described by Concept A. Although representing inheritance relationships is possible in both traditional semantic networks and KL-ONE networks, KL-ONE has a major advantage of having a formal language, while a semantic network usually deals only with graphic representation. Thus, the inference mechanism in KL-ONE is more natural and simpler than in the traditional semantic network.

KL-ONE has contributed to the field of Knowledge Representation in many aspects. These include the "epistemological" primitives, which provide clearer semantics than the traditional semantic network. It also introduced the idea of clear separation between *assertions* and *descriptions* of objects. These contributions have set out a new research framework to overcome some of the limitations in the traditional semantic network representation. KL-ONE has been one of the most influential knowledge representation systems. A number of Knowledge Representation research efforts have been developed based on the KL-ONE framework. These KL-ONE successors are known as the KL-ONE family. A summary of the main features and themes of KL-ONE and its successors can be found in [56].

2.2.7 Description Logic

The ideal characteristics of computational logic include expressiveness, decidable and efficient reasoning and sound and complete inference procedures. However, it has long been known that there is a fundamental tradeoff between expressiveness and the computability of reasoning procedures [45]. FOL has a high degree of expressiveness, which makes inferences undecidable and inefficient. Description Logic (DL)[1] focuses on computability while maintaining a considerable degree of expressiveness. The semantics of DL is based on the structured representation of KL-ONE, which is based on frames. Description logic can be considered as a unifying formalism for structured representation, such as frames, or can be considered as a *structured* fragment of FOL.

[1] Also known as Terminological logic or Concept language

Description logic theory is divided in two parts: TBox and ABox. The "T" in TBox represents "terminological" and "A" in ABox represents "assertional". TBox deals with the definitions of concepts, while ABox deals with assertions over facts. The clear separation between definition and assertion had been introduced in KL-ONE and was later emphasized and formalized into two separate boxes in KRYPTON [57]. TBox allows the establishment of taxonomies of structured terms. Expression in TBox is equivalent to the level of noun phrases in natural language. ABox allows the establishment of descriptive facts about the domains of interest. Expression in ABox is equivalent to the level of sentences in natural language. For example, an expression of "a person with at least 3 children" would go to TBox while an expression of "Every person with at least 3 children owns a car" would go to ABox.

Brachman et al. illustrated the relationship of TBox and ABox as shown in Figure 2.5 [57].

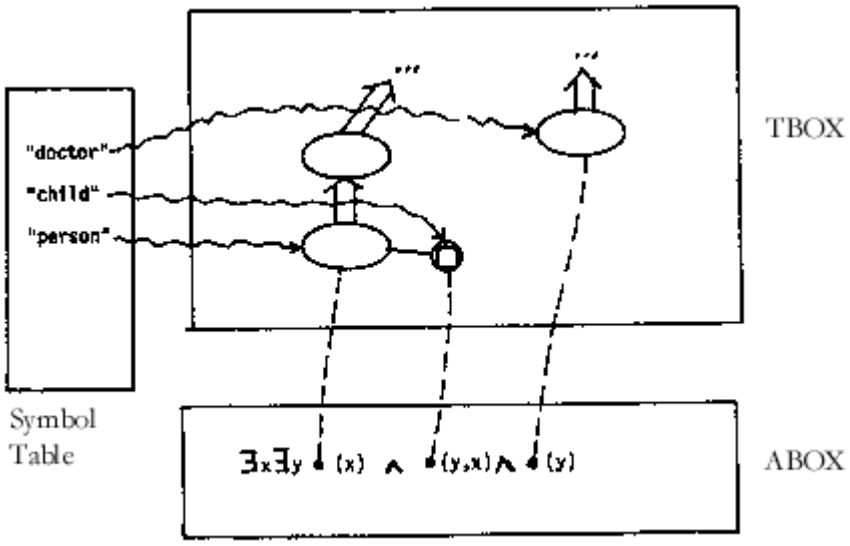


Figure 2.5. TBox, ABox and its relationship

In order to achieve high computability, Brachman & Levesque [46] introduced the reduced version of FOL, which limits the form of expression to frame description form. The logic is known as $\mathcal{FL}^-$. The reasoning service for $\mathcal{FL}^-$, is provided by determining subsumption relationships between concepts. For example, *Student* concept is subsumed by *Person* concept if every member of *Student* concept is also a member of *Person* concept. The computation of this

reasoning task in $\mathcal{FL}^-$ is decidable in polynomial time [46]. The syntax rules of $\mathcal{FL}^-$ are shown in Table 2.2, where C, D are *concepts*, A is a *primitive concept* and R is a *primitive role*.

Table 2.2. Syntax rules of the $\mathcal{FL}^-$ language

C, D →	A |	(primitive concept)
	C ⊓ D |	(conjunction)
	∀R.C |	(universal quantification)
	∃R	(existential quantification)

The only logical connective that $\mathcal{FL}^-$ provides is conjunction of concepts. The value restriction of role is only allowed in universal quantification, not in existential quantification. $\mathcal{FL}^-$ is known as the simplest structural description logic.

While $\mathcal{FL}^-$ provides good computability, its expressiveness is limited. This has led to many variations that add more expressive power of the language while maintaining computability. The $\mathcal{AL}$ language [58] adds more expressive power to the $\mathcal{FL}^-$ language. $\mathcal{ALC}$, a version of $\mathcal{AL}$ language, is one of the simplest propositional DLs. The syntax rule of the $\mathcal{ALC}$ language is shown in Table 2.3.

Table 2.3. Syntax rules of the $\mathcal{ALC}$ language

C, D →	A |	(primitive concept)
	⊤ |	(top concept)
	⊥|	(bottom concept)
	C ⊓ D |	(conjunction)
	C ⊔ D	(disjunction)
	¬ C	(complement)
	∀R.C |	(universal quantification)
	∃R.C	(existential quantification)

2.3 INTEROPERABILITY ON THE WORLD WIDE WEB

The Web was designed with the assumption that the data formats for the Web would proliferate [59]. The Hypertext Transfer Protocol (HTTP) [60], the Web communication protocol, was designed to support various kinds of data formats via the Multipurpose Internet Mail Extensions (MIME) types. Despite the generalized design, the Hypertext Markup Language (HTML) has become the de facto standard for Web documents.

2.3.1 Markup Languages

The idea of a markup "language" was introduced in 1967 by William Tunnicliffe at a Canadian Government Printing Office meeting [61]. The language was called 'generic coding' to distinguish it from 'specific coding' which used control characters or a set of operations (procedural instructions) to instruct software on how to display documents. Generic coding introduced the idea of a declarative markup language. Rather than defining a set of operations, it used descriptive tags to instruct the display software how to format the document. For example, to format a heading of a document, a descriptive "HEAD" tag may be inserted around the heading text.

In 1969, Charles Goldfarb, together with Edward Mosher and Raymond Lorie, at IBM developed the Generalized Markup Language (GML) based on Tunnicliffe's generic coding. GML[1] introduced the idea of allowing users to design document structures using a formally-defined "document type definition". GML played an important role in document publishing projects at IBM through the 1970s.

Further research on GML led, in 1978, to the initiation of a project, supported by the American National Standards Institute (ANSI), focused on the design of a text description language standard. The language was later named the Standard Generalized Markup Language (SGML). The first working draft of the SGML standard was published in 1980. SGML was

[1] The name represents the initials of its three inventors.

accepted by the International Organization for Standardization (ISO) as an ISO standard (ISO 8879) in 1986 [62]. SGML is an international standard for the device-independent, system-independent representation of texts in electronic form.

Three major characteristics of SGML are descriptive markup, document type definition, data representation independence [63].

SGML does not specify procedural instructions, such as indenting text, stepping to the next line, etc. These procedural instructions are usually platform-dependent. Giving explicit procedural instructions would lead to a language that is not device-independent and not interoperable between systems. SGML only describes the logical structure of elements of document. It leaves the interpretation of document formatting to the processing software. Thus, SGML separates the content and structure of the document from its presentation.

The document type concept, which was originated with GML, allows document creators to create their own document type. The logical structure of a *document type*, i.e. book, recipe or brochure, can be defined in a formal definition called a Document Type Definition (DTD). Once a DTD has been defined, a document can be instantiated from it. The document must follow the structural rules specified in its DTD. The ability to create DTDs makes the SGML markup language extensible.

SGML documents are designed to be transportable from one hardware/ software environment to another without loss of information. SGML provides encoding schemes that allow different character sets to be displayed correctly in different environments. This machine-independent encoding allows characters to be transformed or substituted to appropriate characters from system to system.

2.3.2 Hypertext Markup Language (HTML)

In 1989, Tim Berners-Lee and Robert Caillau at the European Organization for Nuclear Research (CERN - Conseil Européen pour la Recherche Nucléaire) collaborated on developing a universal

linked information system for the CERN community. In October of 1990, the system was named the “World-Wide Web". One of the requirements of this system was a formatting language for the hypertext documents. Given the use of SGML by the CERN community, Berners-Lee developed an SGML DTD called the Hypertext Markup Language (HTML). In 1991, he put the code and specifications for HTML on the Internet.

As the World Wide Web became known among the Internet community, HTML was further extended from its original specification. In June 1993, Berners-Lee released an IETF draft version of the Hypertext Markup Language [64]. However, the implementation of HTML by many of the WWW browsers still extended beyond what had been defined in the draft. The first successful attempt in standardizing HTML was HTML 2.0 (IETF RFC1866) [65]. HTML 2.0 attempted to capture the state of HTML as implemented in the WWW browsers as of June 1994. HTML 2.0 was the de facto HTML standard until its replacement by HTML 3.2 [66] in January 1997 and HTML 4.0 [67] in December 1997.

HTML consists of four major components: *tag*, *element*, *attribute* and *link*. The first three components are markup features of SGML. Links were added to enable hypertext. HTML required an addressing scheme for linking WWW resources. Berners-Lee designed a Uniform Resource Identifier (URI) addressing scheme[1] [68] that allowed HTML documents to be linked to other resources on the Web. By combining the descriptive markup capability of SGML with the linking capability, HTML has served as a common format for publishing WWW documents.

Despite its success and popularity, HTML was criticized by the SGML community. HTML, as implemented, lacked *extensibility, structure and validation* [69].

HTML is a simple set of markup tags. HTML does not allow document authors to extend the syntax. Extending HTML would lead to incompatibility. This makes the extensibility of HTML

[1] URI could be location-dependent, i.e. Uniform Resource Locator (URL) or location-independent, i.e. Uniform Resource Name (URN).

low compared to its meta-language, SGML. SGML does not define particular tags but the rules by which an author can create a unique DTD.

HTML also lacks a clear separation between document structure and presentation. While some HTML tags are used to identify document structures, such as <*p*> for paragraph, <*ol*> for ordered list, other tags specify how the text should be displayed, such as <*b*> for bold text, <*font*> for properties of displayed font. This makes the document structure underlying in an HTML document obscure. The lack of structure is problematic when there is a need to represent structured or semistructured data on the Web [70].

The last major limitation of HTML, based primarily on its loosely structure from, is the absence of any validation function. The HTML specification does not specify mechanisms for HTML applications to check the validity of HTML documents. Further, the vast majority of HTML documents available on the Web are not valid HTML documents. One of the reasons is that most of the WWW browsers will display invalid HTML documents, even ones with incorrect syntax. In SGML, validity is important. SGML requires that SGML documents conform to the syntax, i.e., be well-formed, and the structure, i.e. be valid by conforming to a DTD.

While SGML provides a broad solution, it has one major drawback -- it is too complex. Adding SGML processing software to existing WWW browser would require massive changes in browsers. This makes it politically and economically difficult to implement. This led the W3C to design a simplified version markup language of SGML to replace HTML[1].

2.3.3 Extensible Markup Language (XML)

The eXtensible Markup Language (XML) was developed by the XML Working Group, formed under the supervision of World Wide Web Consortium (W3C) in 1996. The XML specification was finally approved as W3C recommendation in February 1998.

[1] However, the final form of XML *and* its companion standards may actually be more complex than SGML.

In some ways, XML is a subset of SGML. One of its goals is "to enable generic SGML to to be served, received, and processed on the Web in the way that is now possible with HTML" [26]. XML is designed to interoperate with SGML. An XML document is also a conforming SGML document. Detailed comparison of XML and SGML can be found in [71].

XML is currently supported in some ways by the major WWW browsers. XML documents can be parsed and displayed in the WWW browsers. The application programing interfaces (APIs) to process XML documents have become publicly available. These includes the Document Object Model (DOM) [72] and Simple API for XML (SAX) [73]. With the availibility of public software modules to process XML documents, XML is being more frequently used in various Internet applications.

A document is "well-formed" XML if it meets the syntax requirements related to tag formation and element nesting. A well-formed XML document is "valid" if it conforms to the structure specified in a DTD. In other word, a *well-formed* XML document is guaranteed to have the proper *syntax* and a *valid* XML document has a structure consistent with a specific DTD. Unlike SGML, a DTD is not mandatory for an XML document unless validity is of concern. This allows XML to be more flexible. An XML document is more restricted than an HTML document, in that well-formedness must be met[1]. Thus, an XML document is syntactically more restricted than an HTML document but more structurally flexible than an SGML document.

While a DTD can define the structure of XML document, the need for better datatype control, along with other needs, led to a replacement for DTD's called Schema [74].

XML Schema[2] [75] offers facilities for describing the structure and constraining the contents of XML documents. It supports data typing for specifying constraints, such as range, precision, etc, of the data. The XML Schema Specification Part 2 [76] has defined primitive data types.

[1] An HTML document does not have to be well-formed

[2] XML Schema became a W3C Recommendation in May 2001

XML Schema also provides mechanisms for document authors to specify complex data types, which is similar to the composition of a data structure in programming languages.

XML Schema also extends DTD functionality by focusing on the reuse and interoperation. XML Schema allows the authors to reuse and integrate existing schemas using the XML namespace facility [77]. A document under SGML and XML may implement one and only one DTD. Using namespaces, a document author can refer to multiple schemas that provide document models that can be mixed in the document. The use of namespace allows the XML schemas to be deployed on a large scale without collisions between elements from different schemas with the same name.

XML documents are intended for use on the WWW, which is a hypertext environment. This requires XML to have an ability to express linking information among XML documents and other Internet resources. To this end, the W3C has defined a specification for expressing links in an XML document. The language is known as XML Linking Language (XLink)[1] [78]. XLink is designed to be more expressive and more powerful than hypertext linking as defined under HTML. Some additional features of XLink include bidirectional links, multiple-destination links, and out-of-line links. The syntax of XLink is specified using XML[2]. An XML link must use an URI [68] to address a resource. An XML link uses the XML Pointer Language (XPointer) specification [79] to identify specific portions in an XML resource as link target.

Like SGML, XML separates document structure from presentation. Stylesheets have been widely used in defining the presentation of a markup document. By attaching stylesheets to structured documents, authors and readers can influence the presentation of documents without interfering with document structure. The Extensible Stylesheet Language (XSL) [80] is a language for expressing stylesheets for XML documents.

---

[1] XLink became a W3C recommendation in June 2001

[2] All of the XML companion standards, using schema and namespaces, are defined in the form of XML documents. The recursive nature of these definitions makes many of these objects much more difficult for human to read, but greatly decreases the complexity of the parser engines that need to be written.

XSL consists of two parts: XML Transformation language (XSLT) [81], and an XML vocabulary for specifying formatting semantics. XSLT language is a language for transforming XML documents into other XML documents. XSLT can also be used independently of XSL formatting objects. XSLT makes use of the expression language defined by XPath [82] for *addressing* and *matching*. XSLT uses XPath to select parts of an XML document for processing. XSLT also provides facilities for string and number manipulation.

2.3.4 Web Resource Identifier

Identifiers are simply names, which are used for identifying things. Identifiers that are *uniform*, i.e. standardized, are important to the communication between parties [83]. For example, the International Standard Book Number (ISBN) has been important to booksellers, publishers and libraries in referring to the printed books. One of the essential attributes of an identifier is its *uniqueness* [84]. A unique identifier must specify one and only one object in the object space. However, this does not imply that an object must have only one identifier.

The need for a uniform identifier scheme for objects in a large and decentralized environment like the Web is inevitable. The Uniform Resource Locator (URL) has served as the unique identifier for the resource on the Web. URL consists of a service name and parameters that passed to the service. For the web resources accessible via the Hypertext Transfer Protocol (HTTP), the service name is "http" and the parameters are a host name and a file name on the host. Although URL is simple and could be easily implemented, it is a location-dependent identifier scheme. When the identified resource is moved or removed, the URL identifier will be no longer valid. As a result, the URL scheme cannot guarantee persistence of the identifiers. This brings a common problem of broken links on the Web. By persistence, the lifespan of the identifiers must not be limited by the lifespan of the objects they identify.

The Uniform Resource Name (URN) has been proposed as a persistent and location-independent identifier for the resources on the Internet. URN is developed as one of the URI schemes, beside the URL and the Uniform Resource Characteristic (URC). In practice, a URN

may also be a URL such as the Persistent Uniform Resource Locator (PURL[1]). The requirements for the URN, specified in RFC 1737 [85], include the guarantee of uniqueness and persistence of the identifiers. The URN scheme utilizes the naming resolution service, which will resolve a URN identifier to a resource location. Thus when the identified resource is moved or removed, the naming resolution service will be responsible for maintaining the uniqueness and persistence of the identifier. Some promising implementations of the URN include the Handle system [86] and the Digital Object Identifier (DOI) [87].

The URN syntax specification has been laid out in RFC 2141 [88], but is still subject to disputes in some areas of interpretation [84]. The URN general syntax is defined as:

<URN> ::= “urn:” <NID> “:” <NSS>

The first component, the string “urn”, indicates that this is a URN. The second component is the namespace identifier (NID), which indicates how the next component should be interpreted. The third component is the namespace specific string (NSS), which is the unique label of the resource within the given NID. Examples of valid URNs are: “urn:isbn:0393041530”, “urn:hdl:cnri.dlib/august95”. It is recommended that the experimental namespaces that are not explicitly registered with the Internet Assigned Numbers Authority (IANA[2]) append the prefix “X-” to the <NID> [89].

2.4 METADATA

Metadata is “data about data”. It is information about a document, such as author, publication date, etc. The International Federation of Library Associations (IFLA) [90] defines metadata as follows:

[1] http://www.purl.org/

[2] http://www.iana.org/

Metadata is data about data. The term refers to any data used to aid the identification, description and location of networked electronic resources.

This definition has limited the scope of metadata usage to electronic resources. Another definition of metadata from Caplan [91] is:

Metadata really is nothing more than data about data; a catalog record is metadata; so is a TEI header, or any other form of description.

According to Caplan, metadata can be used to describe any resources. Caplan does not limit the scope of metadata to electronic resources. Under this definition, a traditional library catalog card is also one kind of metadata. Wool [2] indicates that this definition is preferable as it suggests that metadata is not something new. It has been used for centuries by librarians and publishers. Metadata can be viewed as a kind of cataloging information. The definition of metadata and scope of its usage are still a debate in the library community [92;93].

2.4.1 Dublin Core

In October 1994, at the second International World Wide Web Conference, Stuart Weibel, Senior Research Scientist at the Online Computer Library Center (OCLC), pointed out the need for an agreement on semantics for Internet resources. From a librarian's point of view, this semantics would be equivalent to creating simple catalogue cards for the Internet resources that are not domain-specific but can work across disciplines [94]. This information would help people to describe their materials in order to help Internet users find materials they are looking for.

This initiative led to the OCLC/NCSA Metadata Workshop in March 1995, in Dublin, Ohio, which is also known as the first Dublin Core workshop. The goal was to reach an agreement on a set of simple *metadata elements* that could be used to describe networked digital resources, i.e. resources on the World Wide Web. The scope of the resources that were to be described by these metadata elements, were limited to *document-like objects*, or *DLOs***.** DLOs are primarily text

resources. By restricting the focus to DLOs, the design of metadata elements would resemble the cataloging information that are used to describe traditional print materials.

The result of the agreement was a set of 13 metadata elements, known as the *Dublin Core Metadata Element Set* or *Dublin Core***.** The original Dublin Core elements are listed as the first 13 elements of the Table 2.4 [95].

Table 2.4. Dublin core 1.1 metadata element set

Element Name	Meaning
1. Subject	The topic addressed by the work
2. Title	The name of the object
3. Author	The person(s) primarily responsible for the intellectual content of the object
4. Publisher	The agent or agency responsible for making the object available
5. OtherAgent	The person(s), such as editors and transcribers, who have made other significant intellectual contributions to the work
6. Date	The date of publication
7. ObjectType	The genre of the object, such as novel, poem, or dictionary
8. Form	The physical manifestation of the object, such as Postscript file or Windows executable file
9. Identifier	String or number used to uniquely identify the object
10. Relation	Relationship to other objects
11. Source	Objects, either print or electronic, from which this object is derived, if applicable
12. Language	Language of the intellectual content
13. Coverage	The spatial locations and temporal durations characteristic of the object
14. Contributors	Person(s) who contribute to the content of the resource
15. Rights	Copyright information

The underlying design principles of the Dublin Core are *intrinsicality*, *extensibility*, *syntax independence*, *optionality*, *repeatability*, and *modifiability* [95]. *Intrinsicality* is an ability to describe the resource from its content; no context of use is needed. *Extensibility* is an ability to add extra elements for domain specific information; the core elements must maintain backward-compatibility when they are updated. Dublin core defines the semantics of elements, but not syntax. This makes it usable in a wide range of applications. All the elements in Dublin Core are *optional*. All the elements in Dublin Core are *repeatable*, e.g. to identify multiple authors of the resources, the "author" field can be repeated. Dublin Core allows sophisticated users to use optional qualifiers [96] to specify specific definition of the element (*modifiability*), i.e. "Subject

(scheme=LCSH)" indicating that the subject terms are taken from the Library of Congress Subject Headings.

The number of core elements has been kept low in order to make the standard simple and applicable to a wide-range of resources. Dublin Core 1.1 [5] has defined two more elements: *contributors* and *rights*, which results in the total of 15 core elements in the Dublin Core.

2.4.2 Warwick Framework

A year after the Dublin Workshop, a follow-up workshop was held at the University of Warwick, U.K. The workshop addressed several issues, including the assessment of the one-year experiment with the Dublin Core. Another focus of the workshop was to promote interoperability among different metadata schemes. It sought to address how the Dublin Core could work with other metadata standards. The result of the workshop was an architecture, known as the Warwick Framework [6].

The motivation for the development of the Warwick Framework was to allow interoperability among the existing metadata schemes. Some metadata schemes are general and not domain-specific, e.g. MARC[1], while some are domain-specific, e.g. SAE[2]. Different metadata schemes are also created for different purposes. Some are created for descriptive cataloging purpose. Some are created for legal purposes, e.g. to describe the terms and conditions of use. Some are created for rating the suitability of the content for audiences, e.g. Platform for Internet Content Selection (PICS)[3]. The Dublin Core, which is a general descriptive metadata scheme, was not sufficient to capture all the requirements without incorporating other metadata schemes.

The Warwick Framework was designed to support modularity. For example, a legal organization may want to create a metadata set for describing terms and conditions of use, while

[1] http://www.loc.gov/marc/
[2] http://www.sae.org/
[3] http://www.w3.org/PICS/

librarians might just want to create descriptive cataloging metadata set. The Warwick Framework was designed to support the integration of these pieces. It is also designed to support user selectivity. The architecture allows the users to access to a specific set of metadata in a document, i.e. the parental control software can choose to look at the content rating of the document, while the search engine robots may choose to look at the descriptive cataloging information of the document. The need for selectivity is also necessary when the information that looks like metadata to one program becomes data for another application, e.g. a review of a document can be considered as metadata of the document or a part of the document itself. Under the Warwick Framework, users can choose the metadata elements that are appropriate for their needs.

The Warwick Framework allows different metadata sets to coexist in a document by separating each metadata set into a *package*. These packages are then grouped together in a *container*. This architecture is illustrated in Figure 2.6.

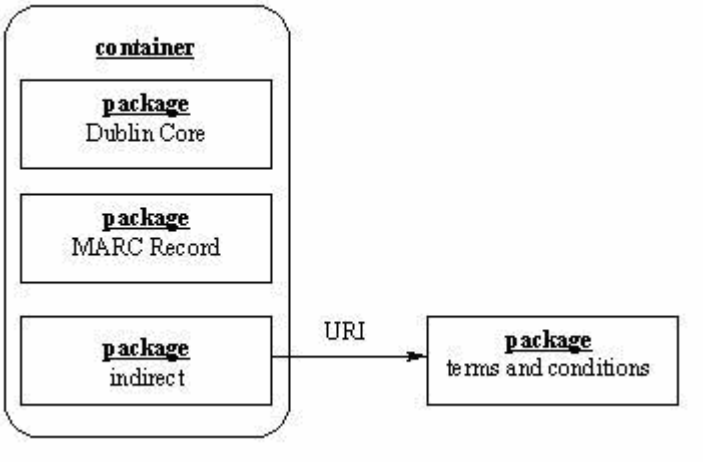


Figure 2.6. The Warwick Framework architecture

Under this architecture, a package can contain records of several metadata sets, such as MARC records, Dublin Core records. A package can also be a link to a package in an external document, i.e. via URL. Under the Warwick Framework, a package can also be a container, making the architecture recursive.

Even though the main architecture of the Warwick Framework has been defined, many problems and issues have been left opened and undefined. These include [97]:

- **Semantic Overlap**: It is possible that two metadata sets have semantic overlap. The Warwick Framework has not defined how the applications should handle this interaction, i.e. how to interpret two metadata records that are conflicting with each other within a single document.
- **Package Type**: In order for a program to interpret the metadata inside a package correctly, it must understand the type of the package, e.g. Dublin Core Package Type or MARC Package Type. As new metadata standards emerge, the architecture needs to specify how the processing software can update its understanding of these new metadata standards.
- **Syntactic interoperability**: The Warwick Framework was syntax-independent. Although this provides flexibility, there needs to be an agreement on the syntax for the metadata sets to interoperate.
- **Efficiency**: The distributed nature of the Warwick Framework can lead to inefficiency, i.e. slow response time, failed connection, etc.
- **Querying**: The selectivity characteristic of the Warwick Framework requires the ability to query and retrieve packages at various levels. The Warwick Framework does not define the metadata querying mechanism.

Although some of the issues are difficult to overcome, the Warwick Framework has set out a framework for interoperability among metadata standards. Many ideas of the Warwick Framework have resulted in the design of the W3C Resource Description Framework (RDF).

2.4.3 Resource Description Framework (RDF)

The Resource Description Framework (RDF) is "*a foundation for processing metadata; it provides interoperability between applications that exchange machine-understandable information on the Web*" [28]. RDF provides a formal data model and syntax for encoding metadata for the purpose of machine processing [7].

RDF may be viewed as an implementation of the Warwick Framework [28]. RDF has proposed some solutions to the problems that were left unsolved in the Warwick Framework.

The Warwick Framework is syntax-independent. Each metadata set can be represented using its own syntax. Also, in the Warwick Framework, there is no unified data model for all metadata sets. Thus, when a new metadata scheme emerges, the processing software will need the information about how to parse the new metadata scheme. In response to these issues, RDF extends the Warwick Framework by defining a unified data model and syntax that all metadata standards can share. RDF also makes use of XML namespace [77] to avoid conflicting definitions of the same term.

Another major influence in the design of RDF comes from Knowledge Representation (KR). RDF is designed to represent metadata for Web resources in a form that could allow for computer programs to make use of in an intelligent manner [98].

The RDF specifications were released in two parts: the *Resource Description Framework (RDF) Model and Syntax Specification*[1] [7] and the *Resource Description Framework (RDF) Schema Specification 1.0* [28]. The follows provide a summary of the specifications.

2.4.3.1 RDF Data structure The RDF data structure can be introduced by considering a simple example from the RDF specification. Giving the following statement about a resource:

***Ora Lassila** is the **creator** of the resource **http://www.w3.org/Home/Lassila**.*

In RDF, the statement is considered to contain the following elements with the following values:

Elements	***Value***
Subject (Resource)	*http://www.w3.org/Home/Lassila*
Predicate (Property)	*Creator*
Object (literal)	*"Ora Lassila"*

[1] RDF Model and Syntax Specification became a W3C recommendation in February 1999.

The statement can be simply represented in the form of the triple "*Predicate (Subject, Object)"*

```
Creator (http://www.w3.org/Home/Lassila, "Ora Lassila")
```

The *Subject* and *Predicate* elements can be represented using URIs, while *Object* can be represented using URIs or literal string.

The triple can also be represented graphically as a directed label graph (DLG) shown in Figure 2.7.

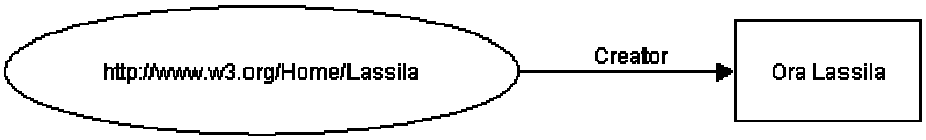


Figure 2.7. A basic statement in RDF

The oval represents a resource identified by a URI and the rectangle represents a literal string. The arrow represents a property of the resource.

The basic RDF statement provides a mechanism for representing descriptions of Web resources in the form of property-value pairs.

RDF also provides a mechanism for document authors to make a statement about multiple resources, such as a list of authors, a list of documents. RDF has defined the notion of *container* to allow the statement about a collection of resources or collection of strings. Container can be one of the following three types.

- *Bag*: used when the order of the items in the collection is not important and duplicates are allowed.
- *Sequence*: used when the order of the items in the collection is important and duplicates are allowed.

- *Alternative*: used when any one of the items in the collection can be picked, i.e. list of Internet mirror sites

RDF always allows duplication of metadata statements. Thus the notion of *Set* (unordered list without duplication) is not defined in RDF.

RDF also allows document authors to make statements about statements. This mechanism is called *reification*. Reification is a mechanism of transforming a statement into a resource. This will allow document authors to make the statements about it. RDF allows a statement to be explicitly constructed as a resource using four properties: *rdf: subject*, *rdf: predicate*, *rdf:object* and *rdf:type*. An example of a reified statement can be shown as follows:

Given the following example statement from the RDF specification:

> ***Ralph Swick*** *says that* ***Ora Lassila*** *is the* ***creator*** *of the resource* ***http://www.w3.org/Home/Lassila****.*

The statement can be represented in an RDF graph as shown in Figure 2.8:

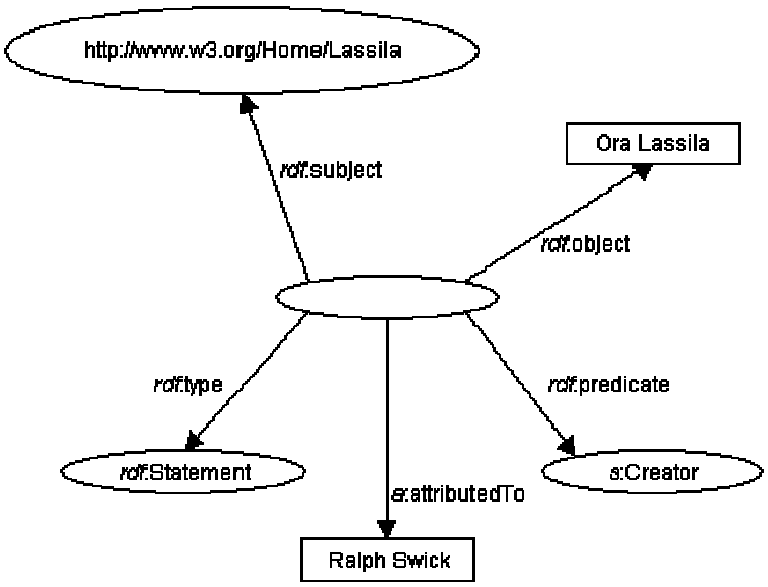


Figure 2.8. A reified statement in RDF

The blank oval represents an anonymous resource. The resource is described by five properties: four of which are about the elements of the statement (subject, predicate, object, type) and another one states that the statement is cited by (attributedTo) a person ("Ralph Swick").

2.4.3.2 RDF Syntax A serialization syntax allows the creation and exchange of metadata information. Although RDF is independent of syntax, the designers of RDF have chosen XML as the default syntax for RDF due to its strength as universal data interchange format. Another main reason is that it allows RDF to use the namespace facility of XML. The use of the XML namespace facility in RDF helps to avoid the confusion and conflict in referencing terms.

The following XML namespace declaration associates namespace prefix, *"rdf"*, with the URI of schema for RDF, *http://www.w3.org/1999/02/22-rdf-syntax-ns#.*

```
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

The RDF data can be represented in two kinds of XML syntax: *full serialization syntax* and *abbreviated syntax*.

Given the RDF data model of the sentence,

Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila.

The full RDF's serialized syntax in XML is:

```
<rdf:RDF>
  <rdf:Description about="http://www.w3.org/Home/Lassila">
    <s:Creator>Ora Lassila</s:Creator>
  </rdf:Description>
</rdf:RDF>
```

where the prefix 's' refers to an XML namespace declaration of the schema, which defines the metadata terms, such as

```
xmlns:s="http://description.org/schema#"
```

For the purpose of compactness, RDF allows the syntax to be written in *abbreviated syntax* form. The same example can be written in the abbreviated form as follows:

```
<rdf:RDF>
  <rdf:Description about="http://www.w3.org/Home/Lassila"
                   s:Creator="Ora Lassila" />
</rdf:RDF>
```

One benefit of abbreviated syntax is that it hides RDF data in an XML or HTML document.

2.4.3.3 RDF Schema Under RDF, a schema is used to define terms, as well as restrict terms usage. It should be noted that RDF Schema should not be confused with the XML Schema. RDF schema is simply a machine-readable dictionary. For example, in a metadata application, an RDF schema declares the vocabulary of the metadata elements and their corresponding meanings. These meanings are described in term of the relationships between terms.

In order to allow the creation of RDF schema in a uniform way, W3C has released the RDF Schema Specification 1.0[1] [28], as a separate specification from the RDF Model and Syntax specification [7]. The specification does not specify vocabulary for metadata elements, i.e. "creator", "subject". It only provides a language, *RDF schema specification language*, for the creation of metadata elements. The language itself is a meta-language in that it is a language used to create RDF Schema.

The following XML namespace declaration associates namespace prefix, *"rdfs"*, with the URI of RDF Schema for the RDF Schema specification language.

```
xmlns:rdfs=" http://www.w3.org/2000/01/rdf-schema#"
```

Class and property The RDF schema specification language allows the semantics of new vocabulary to be expressed in terms of relationships to other vocabularies. The relationship can be expressed in terms of relation to an existing class, defined by *rdfs:Class*, or existing property,

[1] RDF Schema specification became a W3C recommendation in February 2004.

defined by *rdf:Property*. For example, *rdfs:subClassof* defines an inheritance relationship between two *classes*; *rdfs:subPropertyof* defines an inheritance relationship between two properties; *rdf:type* defines instance-of relationship between a resource and a class. RDF Schema also allows the definitions of vocabulary to be defined elsewhere. The references to external definitions can be defined using *rdfs:seeAlso*.

Constraints A vocabulary can also be defined in terms of constraints and restrictions. The RDF Schema specification language provides *rdfs:range* and *rdfs:domain* elements for expressing constraints. The *rdfs:range* element is used to specify the value restriction of a property. For example, the following RDF statement states that the value of "author" property must be a resource of "Person" class, where "Person" is defined elsewhere in the same document.

```
<rdf:Description rdf:ID="author">
    <rdfs:range rdf:resource="#Person"/>
</rdf>
```

The *rdfs:domain* element is used to specify the class where the property can be applied to. For example, the following RDF statements state that the "author" property can be used with "Book" class, where "Book" is defined elsewhere in the same document.

```
<rdf:Description rdf:ID="author">
    <rdfs:domain rdf:resource="#Book"/>
    <rdfs:range rdf:resource="#Person"/>
<rdf>
```

2.5 ONTOLOGY ON THE WORLD WIDE WEB

According to McGuinness [99], **language** and **environment** are two major concerns in the deployment of ontology. Language is crucial to how the ontology is created. Environment is crucial to how the ontology is maintained and used over time.

2.5.1 Ontology Language for the World Wide Web

According to van Harmelen [100], some properties of a good ontology language include:

- be expressive enough to capture many ontologies;
- have a common syntax and should be easy to integrate with ontologies that are created in other languages;
- have formal semantics such that machines can understand and reason on it.

In a way, RDF Schema specification language could be considered a simple ontology language for the Web. In particular, RDF Schema specification language provides basic primitives for modeling ontology, such as class, property, subclass-of, subproperty-of, domain and range.

However, as an ontology language, the RDF Schema specification language has been considered insufficient in terms of its expressive power [101;102]. This expressiveness inadequacy includes the lack of logical connectives such as conjunction, disjunction, and negation in RDF Schema. Further, RDF Schema does not allow one to define the property of a property. For example, one cannot define that a property is transitive (a(x,y), a(y,z) → a(x,z)) or symmetric (a(x,y) = a(y,x)). These become limitations in some applications which require the use of expressive ontology. Another major inadequacy of RDF Schema specification language is its lack of inference mechanism, which is crucial to automated processing by computer programs.

The insufficiency of the RDF Schema specification language has led to the efforts in designing the ontology languages with more expressive power for creating and sharing ontologies on the Web. These languages usually offer more expressive power than the RDF

Schema specification language. The formal semantics of these languages are usually defined using some forms of logic.

The following sections review four ontology languages designed for the Web. Although the focus will be on the DARPA Agent Markup Language (DAML) and the Web Ontology Language (OWL), descriptions of the Simple HTML Ontology Extension (SHOE) and the Ontology Inference Layer (OIL) provide additional background.

2.5.2 Simple HTML Ontology Extension (SHOE)

One attempt to define an expressive ontology language is SHOE (Simple HTML Ontology Extension). SHOE is a knowledge representation language that allows web pages to be annotated with ontology-based semantics [103]. SHOE has been proposed as an extension to HTML. SHOE was developed at the University of Maryland at College Park in 1996 [104], prior to the development of XML and RDF. The syntax of SHOE is defined in a DTD (initially an SGML DTD and later an XML DTD [105]).

SHOE separates the terminological descriptions, known as *ontology* part, from the assertions, known as *instance* part. The ontology part in SHOE allows one to define *Category* definitions. SHOE ontology also allows one to define *Relation* definitions. A Relation in SHOE can be an *n*-ary predicate. SHOE also allows inference rules to be defined in the ontology specification in the form of horn clause, i.e. $a \wedge b \wedge c \Rightarrow d$. An example of SHOE ontology definition is shown in the following example [103].

Table 2.5. SHOE ontology example

```
<HTML>
<HEAD>
   <TITLE>University Ontology</TITLE>
Tell agents that we're using SHOE
   <META HTTP-EQUIV="SHOE" CONTENT="VERSION=1.0">
</HEAD>
<BODY>
Declare an ontology called "university-ontology".
   <ONTOLOGY ID="university-ontology" VERSION="1.0">
Borrow some elements from an existing ontology, prefixed with a "b."
          <USE-ONTOLOGY ID="base-ontology" VERSION="1.0" PREFIX="b"
                 URL="http://www.cs.umd.edu/projects/plus/SHOE/base.html">
Define some categories and subcategory relationships
          <DEF-CATEGORY NAME="Person" ISA="b.SHOEentity">
          <DEF-CATEGORY NAME="Organization" ISA="b.SHOEentity">
          <DEF-CATEGORY NAME="Worker" ISA="Person">
          <DEF-CATEGORY NAME="Advisor" ISA="Worker">
          <DEF-CATEGORY NAME="Student" ISA="Person">
          <DEF-CATEGORY NAME="GraduateStudent" ISA="Student Worker">
Define some relations; these examples are binary, but relations can be n-ary
          <DEF-RELATION NAME="advises">
                 <DEF-ARG POS=1 TYPE="Advisor">
                 <DEF-ARG POS=2 TYPE="GraduateStudent"></DEF-RELATION>
          <DEF-RELATION "age">
                 <DEF-ARG POS=1 TYPE="Person">
                 <DEF-ARG POS=2 TYPE="b.NUMBER"></DEF-RELATION>
          <DEF-RELATION "suborganization">
                 <DEF-ARG POS=1 TYPE="Organization">
                 <DEF-ARG POS=2 TYPE="Organization"></DEF-RELATION>
          <DEF-RELATION "works-for">
                 <DEF-ARG POS=1 TYPE="Person">
                 <DEF-ARG POS=2 TYPE="Organization"></DEF-RELATION>
Define a transfers-through inference over working for organizations
          <DEF-INFERENCE>
                 <INF-IF>
                        <RELATION NAME="works-for">
                               <ARG POS=1 VALUE="x" VAR>
                               <ARG POS=2 VALUE="y" VAR></RELATION>
                        <RELATION NAME="suborganization">
                               <ARG POS=1 VALUE="y" VAR>
                               <ARG POS=1 VALUE="z" VAR></RELATION></INF-IF>
                 <INF-THEN>
                        <RELATION NAME="works-for">
                               <ARG POS=1 VALUE="x" VAR>
                               <ARG POS=2 VALUE="z" VAR></RELATION></INF-THEN>
          </DEF-INFERENCE>
   </ONTOLOGY>
</BODY>
</HTML>
```

In the university ontology example, five categories are defined: *Person*, *Organization*, *Worker*, *Advisor*, *Student* and *GraduateStudent*. These categories are defined in terms of their relationships to each other and to the base entities (defined in the SHOE base-ontology). The example ontology also defines four relations: *advises*, *age*, *suborganization* and *works-for*. These relations are defined in terms of their value restrictions, which can be either in the form of allowed categories or allowed data types. SHOE supports four basic types: strings, numbers, dates and booleans. Inference rules can also be defined in the ontology. From the example

ontology, the inference rule defined is equivalent to the following FOL sentence: ($\forall x \in$ Worker) ($\forall y \in$ Organization) ($\forall z \in$ Organization) works-for(x,y) $\wedge$ suborganization(y,z) $\Rightarrow$ works-for(x,z).

Once the meanings of the categories and relations and their relationships are defined in the ontology, one can use them to make claims or assertions in a web page. The claims can be *category* claims, such as a claim that Mike is a graduate student, where graduate student *(GraduateStudent)* is defined by the conjunction of *student* and *worker* categories. The web page may also make a *relation* claim that Mike is advised by John, where *advise* is a relation of *Advisor* and *GraduateStudent*. This is shown in the following SHOE instance example [103].

Table 2.6. SHOE instance example

```
     <HTML>
     <HEAD>
        <TITLE>John's Web Page</TITLE>
     Tell agents that we're using SHOE
        <META HTTP-EQUIV="SHOE" CONTENT="VERSION=1.0">
     </HEAD>
     <BODY>
        <P>This is my home page, and I've got some SHOE data on it about me and my
advisor. Hi, Mom!</P>
     Create an Instance. There's only one instance on this web page, so we might as well
use the web page's URL as its key. If there were more than one instance, perhaps the
instances might have keys of the form http://univ.edu/john#FOO
        <INSTANCE KEY="http://univ.edu/john">
     Use the semantics from the ontology \university-ontology", prexed with a \u."
        <USE-ONTOLOGY       ID="university-ontology"       VERSION="1.0"       PREFIX="u"
URL="http://univ.edu/ontology">
     Claim some categories for me and others.
        <CATEGORY NAME="u.GraduateStudent">
        <CATEGORY NAME="u.Advisor" FOR="http://univ.edu/mike">
     Claim some relationships about me and others. \me" is a keyword for the enclosing
instance.
        <RELATION NAME="u.advises">
                <ARG POS=1 VALUE="http://univ.edu/mike">
                <ARG POS=2 VALUE=me> </RELATION>
        <RELATION NAME="u.age">
                <ARG POS=1 VALUE=me>
                <ARG POS=2 VALUE="32"> </RELATION>
        </INSTANCE>
     </BODY>
     </HTML>
```

SHOE is one of a few ontology languages that focuses on the consistency of assertions [106;107]. SHOE prevents contradictions by allowing no retractions of knowledge from the knowledge base. SHOE does not allow negation in the claim statement. SHOE also includes the identification of the claimer along with the claim statements such that one can identify who has made the false claims. The consistency of the ontology is also maintained by a versioning

mechanism. Each ontology must have the version number associated with it. Different versions of the same ontology must be in separate files.

2.5.3 Ontology Inference Layer (OIL)

The Ontology Inference Layer[1] (OIL) is a proposed representation and inference language for ontology on the Web. OIL proposes to define an additional layer that provides formal semantic and reasoning capability on top of the RDF Schema layer [102;108]. OIL is designed to be an extensible standard. To achieve this, OIL uses the layered approach. The lowest level of the OIL standard, known as Core OIL, is compatible with RDF Schema specification, except for the RDF reification mechanism. Ontologies defined by the Core OIL language are interpretable by an RDF Schema aware application. The next layer, Standard OIL, adds more features to Core OIL which makes it only partially understood by an RDF Schema aware application. Standard OIL is designed such that it can provide adequate expressive power as well as reasoning support. Instance OIL includes full integration of individuals (instances) into the language. The layers are illustrated in Figure 2.9 [109]. An example OIL ontology is provided in Table 2.7 [110].

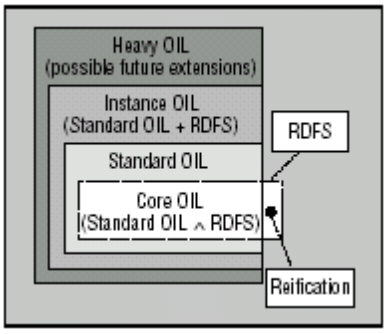


Figure 2.9. OIL's layered language model

[1] Also known as the Ontology Inference Language

Table 2.7. An OIL ontology example

```
1       ontology-container
2               title "African animals"
3               creator "Ian Horrocks"
4          subject "animal, food, vegetarians"
5          description "A didactic example ontology describing African animals"
6          description.release "1.01"
7          publisher "I. Horrocks"
8          type "ontology"
9          format "pseudo-xml"
10         format "pdf"
11         identifier "http://www.cs.vu.nl/~dieter/oil/TR/oil.pdf"
12         source "http://www.africa.com/nature/animals.html"
13         language "OIL"
14         language "en-uk"
15         relation.hasPart "http://www.ontosRus.com/animals/jungle.onto"
16      ontology-definitions
17              slot-def eats
18              inverse is-eaten-by
19              slot-def has-part
20              inverse is-part-of
21         properties transitive
22              class-def animal
23         class-def plant
24              subclass-of NOT animal
25              class-def tree
26              subclass-of plant
27              class-def branch
28              slot-constraint is-part-of
29         has-value tree
30              class-def leaf
31              slot-constraint is-part-of
32              has-value branch
33              class-def defined carnivore
34              subclass-of animal
35              slot-constraint eats
36              value-type animal
37              class-def defined herbivore
38              subclass-of animal
39              slot-constraint eats
40              value-type plant OR (slot-constraint is-part-of has-value plant)
41              class-def herbivore
42              subclass-of NOT carnivore
43              class-def giraffe
44              subclass-of animal
45              slot-constraint eats
46              value-type leaf
47              class-def lion
48              subclass-of animal
49              slot-constraint eats
50              value-type herbivore
51              class-def tasty-plant
52              subclass-of plant
53              slot-constraint eaten-by
54              has-value herbivore, carnivore
```

An OIL ontology consists of two major parts: the *ontology container* and the *ontology definition*. The ontology container provides metadata information for the ontology, such as the title of the ontology, the author of the ontology, etc. OIL uses the Dublin Core Metadata Element Set, Version 1.1 for describing the metadata information. The ontology definition consists of a

set of expressions that describe classes and slots. The ontology definition provides the following kinds of expressions: *class definition*, *slot constraints* and *slot definition*.

Class definition (class-def) associates a class name with its description. Class definitions are *primitive* or *defined*. A class is primitive type if it can be necessarily defined in terms of other classes but not vice versa. From the animal ontology example, lion is a primitive class because it can be necessarily defined as animal but not vise versa (lion $\Rightarrow$ animal). Herbivore, however, is a defined class because it can be necessarily and sufficiently described in term of animal, whose "*eat*" slots must all be filled in by one of plant type (herbivore $\Leftrightarrow$ animal $\wedge\forall$eats.plant). In OIL, a class has primitive type by default. A class definition can be defined in terms of a set of class expressions. A class expression can be expressed as its subclass relationship or its slot constraint. For example, the definition of Herbivore class is shown in lines 37-42 of Table 2.7.

Slot constraint (slot-constraint) can be defined by one of the following: *has-value*, *value-type*, *max-cardinality*, *min-cardinality,* and *cardinality*. The *has-value* constraint is equivalent to the existential quantifier of role in description logic ($\exists$R.C). The *value-type* constraint is analogous to the universal quantifier of role in description logic ($\forall$R.C). The *max-cardinality* and *min-cardinality* give the specific number of instances that are allowed for the slot. The *cardinality* is used when the *min-cardinality* and *max-cardinality* are the same. The cardinality expressions in OIL are similar to number restrictions in description logic. The current version of OIL allows two basic data types: integer and string.

Slot definition (slot–def) associates a slot name with its description. A slot definition can include the following components: *subslot-of*, *domain*, *range*, *inverse* and *properties* (transitive or symmetric). Compared to RDF Schema specification language, *oil:subslot-of* is equivalent to *rdfs:subPropertyOf*. The *domain* and *range* elements in OIL have the same meaning as *rdfs:domain* and *rdfs:range* respectively. The *inverse* and *properties* elements have no equivalent in RDF Schema specification language. *Inverse* allows definition of the slot as having an inverse relationship with other slots. In the animal ontology example, the slot "is-eaten-by" is defined as the inverse of the slot "eats". An inverse is shown in lines 17-18 of Table 5. Properties of slots in OIL can be defined. A slot can be transitive (a(x,y) $\wedge$ a(y,z) $\Rightarrow$ a(x,z)) or symmetric (a(x,y) $\Rightarrow$

a(y,x)). For example, one might define the slot "*longer-than*" as transitive (longer-than(x,y) $\wedge$ longer-than(y,z) $\Rightarrow$ longer-than(x,z)) while defining the slot "*live-with*" as symmetric (live-with(x,y) $\Rightarrow$ live-with(y,x).

OIL provides formal and clear semantics for the ontology language by mapping the OIL expressions to description logic. The formal logic used in OIL is an extension of the $\mathcal{ALC}$ language, known as $\mathcal{SHIQ}$. The letter $\mathcal{S}$ in $\mathcal{SHIQ}$ is shorthand for the $\mathcal{ALC}_{\mathcal{R}}$+ language, which is an extension of the $\mathcal{ALC}$ language that includes Role transitivity. $\mathcal{SHIQ}$ extends the $\mathcal{ALC}_{\mathcal{R}}$+ language by adding a hierarchy of roles ($\mathcal{H}$), inverse roles ($\mathcal{I}$) and fully qualified number restrictions ($\mathcal{Q}$) [110]. In order to enable support for concrete data types such as integer and string, $\mathcal{SHIQ}$ has been extended to $\mathcal{SHIQ}(\mathcal{d})$. It is claimed that $\mathcal{SHIQ}(\mathcal{d})$ can capture the semantics of both Standard OIL and Instance OIL. A complete mapping of OIL language to $\mathcal{SHIQ}(\mathcal{d})$ description logic can be found in [111].

Even though $\mathcal{SHIQ}$ can provide an efficient reasoning service, its lack of support for expressing instance (individual) in a class expression has been a major limitation of the expressiveness of the ontology language. There are many cases where expressing class definition in terms of instance is useful. For example, one might want to define the class of "Italian" as "person" who was born in "Italy" [112]. In this case, Italy is an instance of "Country" class. This cannot be expressed in the ontology language that has no support for instance in a class expression. $\mathcal{SHIQ}$ logic does not support this kind of expression, thus adding this form of expression to OIL ontology language would result in no mapping to description logic. As a result, there would be no reasoning support from the description logic. To overcome this limitation, $\mathcal{SHOQ}(\mathcal{D})$ has been proposed [112] as a formal logic for OIL in place of $\mathcal{SHIQ}$.

$\mathcal{SHOQ}(\mathcal{D})$ is an extension of $\mathcal{SHQ}$ ($\mathcal{SHIQ}$ without support for inverse roles). $\mathcal{SHOQ}(\mathcal{D})$ extends $\mathcal{SHQ}$ by allowing instance in class definition or named individual ($\mathcal{O}$) and has support for concrete data types ($\mathcal{D}$). The reason that $\mathcal{SHOQ}(\mathcal{D})$ extends $\mathcal{SHQ}$, rather than $\mathcal{SHIQ}$, is that

reasoning with inverse roles is known to be difficult or even intractable when combined with either concrete data types or named individuals [112].

2.5.4 DARPA Agent Markup Language (DAML)

The DARPA Agent Markup Language (DAML)[1] was built on W3C XML and RDF, OIL, SHOE, and related efforts [113]. The purpose was to define a unified framework for a Web ontology language based on the existing Web ontology language efforts. DAML released its first ontology language specification, DAML-ONT, in October, 2000 [114]. In December, 2000, DAML+OIL [115] had been released to replace DAML-ONT. DAML+OIL provides clearer semantics while making the language more consistent with the OIL project. This specification was later replaced by the version of DAML+OIL released in March, 2001 [116].

DAML+OIL (March 2001) is divided into two parts. The first part, called the *object domain*, consists of objects that are members of classes defined in the DAML ontology. The latter part is called the *datatype domain*, which consists of values that belong to XML Schema data types. For example, in DAML+OIL, instances of class, e.g. the person "John Smith", would be interpreted separately from instances of data types, e.g. the integer 5. By separating data types from classes, the data types will be modeled outside the ontology language [117]. This helps in maintaining the simplicity and compactness of the ontology language. Separation of data types outside the ontology language also keeps reasoning support for the ontology language implementable.

A brief description of all major DAML elements can be described as follows. For brevity, the explanation focuses on the explanation of the meaning of each DAML+OIL element rather than its syntax rule. A complete reference of all DAML+OIL elements can be found in the DAML+OIL reference description [118]. The following XML namespace declaration associates namespace prefix, *"daml"*, with the URI of RDF Schema for DAML+OIL (March 2001).

```
xmlns:daml=" http://www.daml.org/2001/03/daml+oil.daml"
```

[1] http://www.daml.org/

DAML allows the following forms of expressions: *class element*, *class expression* and *property element*.

Class element associates a class name with its definition. Class definition is defined in terms of the following five optional elements*:* *daml:subClassOf*, *daml:disjointWith*, *daml:disjointUnionOf* , *daml:sameClassAs* and *daml:equivalentTo*. The *daml:subClassOf* allows the definition of a class to be defined in term of its subclass relationship to other classes. The *daml:disjointWith* allows class definition to be defined in term of its complement relationship to other classes. For example, "male" could be defined as a disjoint class of "female". The *daml:disjointUnionOf* allows class definition to be defined in term of the union of disjoint classes. For example, human is a union of male and female, where male and female are disjointed classes. The *daml:sameClassAs* and *daml:equivalentTo*, in the context of class definition, share the same meaning of defining equivalence of classes.

Class expressions allow the construction of a class definition. Class expression can be expressed in the form of one of the following: *a class name*, *an enumeration*, *property-restriction* or *their boolean combination*.

A class name is the name of the class whose definition may be defined. There are two predefined class names: *daml:Thing* and *daml:Nothing*. Every class is a subclass of *daml:Thing*, while *daml:Nothing* is a subclass of every class. From the instance viewpoint, every object is a member of *daml:Thing* and no object is a member of *daml:Nothing*.

Enumeration is expressed by a *daml:oneOf* element followed by a list of enumerated instances. For example, the "Continent" class can be expressed as one of the enumerated list: Europe, Asia, Africa, NorthAmerica, SouthAmerica, and Australia.

Class expression can also be expressed in term of property restrictions, using *daml:Restriction* and *daml:onProperty.* There are two kinds of property restrictions: *ObjectRestriction,* where the property must be an instance from the specified class, and

DatatypeRestriction, where the property must have its value in the specified data type. The restriction can be expressed by one of the following elements: *daml:toClass*, *daml:hasClass* and *daml:hasValue*. The *daml:toClass* is analogous to the universal quantifier in predicate logic. The *daml:hasClass* is analogous to the existential quantifier in predicate logic. The *daml:hasValue* specified the specific instance of class or data type value that is allowed to fill in the property.

The property restriction can also be expressed in term of its cardinality restrictions: *daml:maxCardinality*, *daml:minCardinality* and *daml:cardinality*. The *daml:maxCardinality* and *daml:minCardinality* elements specify the maximum and minimum number of instances to be filled in the specified property. The *daml:cardinality* is the shorthand that is used when the number specified in *daml:maxCardinality* equals *daml:minCardinality*. The expression of cardinality restriction could be expressed using *daml:maxCardinalityQ*, *daml:minCardinalityQ* or *daml:cardinalityQ* respectively.

DAML also allows the combination of class expressions to form a new class expression using the logical connectives. The connectives are expressed by one of the following elements: *daml:intersectionOf*, *daml:unionOf*, and *daml:complementOf*. The *daml:intersectionOf* is analogous to the logical conjunctive operator (AND). The *daml:unionOf* is analogous to the logical disjunctive operator (OR). The *daml:complementOf* is analogous to the logical negation operator (NOT).

Property element associates a property name with its definition. Property definition can be defined in terms of the following elements: *daml:subPropertyOf*, *daml:domain*, *daml:range*, *daml:samePropertyAs*, *daml:equivalentTo* and *daml:inverseOf*. The *daml:subPropertyOf* element defines the property definition in term of its relationship to other properties. The *daml:domain* defines the property in term of what classes it can be applied to. The *daml:range* defines the property in terms of its allowable value. The *daml:samePropertyAs* and *daml:equivalentTo*, in the context of property definition, share the same meaning of stating the equivalence of one property to another property. The *daml:inverseOf* element defines the property in term of its inverse relationship with another property.

There are two major types of properties: *daml:ObjectProperty*, and *daml:DatatypeProperty*. The ObjectProperty defines the property in term of its relationship to objects, while the DatatypeProperty defines the property in terms of its relationship to a data type value. There are three other kinds of properties which represent their special characteristics: *daml:TransitiveProperty*, *daml:UniqueProperty* and *daml:UnambiguousProperty*. A property "P" is defined as *daml:TransitiveProperty* if P(x,y) and P(y,z) imply P(x,z). The *daml:UniqueProperty* is a shorthand notation for the property that has its *maxCardinality* restriction of one. The property that is defined as *daml:UnambiguousProperty* is an inverse of the property that is defined as *daml:UniqueProperty*.

DAML+OIL also provides *daml:sameIndividualAs* and *daml:differenctIndividualFrom* elements for stating that two individuals are the same or different respectively.

In order to allow the expression of a collection of items, DAML+OIL provides the *daml:collection* element as an extension of *rdf:parseType*. The *daml:collection* is meant to be interpreted as an unordered list, aka a bag.

2.5.5 Web Ontology Language (OWL)

The Web Ontology Language (OWL)[1] is a revision of the DAML+OIL ontology language [119]. OWL aims to provide different level of expressiveness support for different needs of applications and tools. It has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full. OWL Lite supports the needs in creating ontology involving simple constraints and taxonomies. OWL DL provides an increasing expressiveness that can gain an efficient inference support by description logic. OWL Full provides the maximum expressivness with no guarantee of efficient inference support.

OWL Lite include the expressiveness provided in the RDF schema specification language such as *Class*, *rdf:property*, *rdfs:subClassOf*, *rdfs:subPropertyOf*, *rdfs:domain*, *rdfs:range* and

[1] OWL specifications became W3C recommendations in February 2004.

Individual. It also provides the expressiveness in stating equality and inequality such as *sameClassAs*, *samePropertyAs*, *sameIndividualAs* and *differentIndividualFrom*. The expressiveness for property characteristics also includes *inverseOf*, *TransitiveProperty*, *SymmetricProperty* and *FunctionalProperty*. The expressiveness for the universal and existential quantifiers is provided in terms of *allValuesFrom* and *someValuesFrom* respectively. The expressiveness for cardinality is provided in terms of *minCardinality*, *maxCardinality* and *cardinality*. However, OWL Lite limits the cardinality values to only zero and one.

OWL DL and OWL Full include the expressiveness provided in OWL Lite plus some additional expressiveness provided. Although OWL DL and OWL Full use the same vocabulary, the expressiveness in OWL DL is more limited. For example, in OWL DL, a class can not be defined as an instance of another class, i.e. individual. In OWL Full, a class can also be defined as a collection of individuals or as an individual. The additional expressiveness provided in OWL DL and OWL Full also include *oneOf*, *disjointWith*, *unionOf*, *complementOf*, *intersectionOf* and the cardinalities whose values can be in any number.

OWL and DAML+OIL are closely related in terms of their design, motivation and applications. The Web ontology language will allow the creation and sharing of ontologies that can be distributed across many systems in a way that is compatible with Web standards. A large number of tools and applications has been developed utilizing such a means. For example, the OWL-based Web Service Ontology (OWL-S[1]), which was defined based on OWL, defines a set of constructs for describing the properties and capabilities of Web services. The created ontologies can be used to facilitate the automation of Web service tasks including automated discovery and execution. Some example ontologies created using OWL-S include those defined based on the Amazon.com Web Services, currency converter Web services[2].

[1] formerly the DAML Services (DAML-S) – http://www.daml.org/services/

[2] These ontologies can be viewed at http://www.daml.org/services/owl-s/examples.html

2.6 CONCLUSION

This chapter reviews and describes some theoretical foundations and the development of the standards for the Semantic Web.The next chapter will discuss the design and development of a system that can be used to support the organization and discovery of information resources on the Semantic Web.

3.0 IMPLEMENTATION

This chapter discusses the implementation of a system. The system is a deduction system that can be used to process the Semantic Web data. The chapter begins with the overview of the system architecture. The description of each system component is subsequently provided.

3.1 SYSTEM ARCHITECTURE

The system architecture of a deduction system for the Semantic Web consists of the three major components: *Information Acquisition*, *Knowledge Base* and *Knowledge Retrieval* components.

The *Information Acquisition* component gathers the Semantic Web data available in the RDF format. It transforms the data into a form that is suitable for the knowledge base. The *Knowledge Base* allows automated deduction to be made over the acquired information. The *Knowledge Retrieval* component allows the gathered information and new conclusions produced by the knowledge base to be retrieved and utilized. The system architecture of the deduction system is summarized in Figure 3.1.

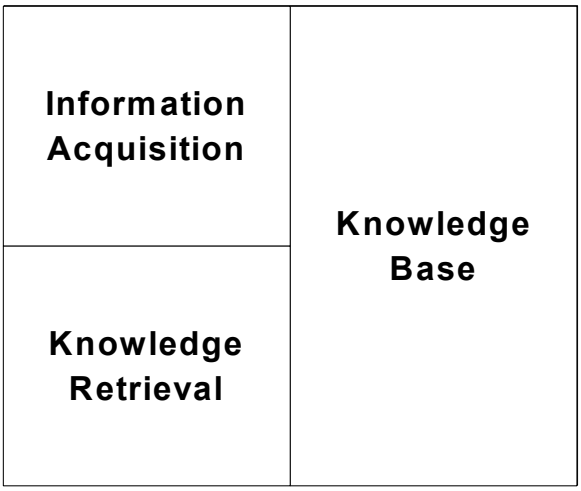


Figure 3.1. System architecture of the deduction system

3.2 INFORMATION ACQUISITION

The major task of the Information Acquisition component is to gather the Semantic Web data in the RDF format and transform them into the Description Logic syntax to be processed by the knowledge base. Figure 3.2 illustrates the information flow in acquiring the information into the deduction system.

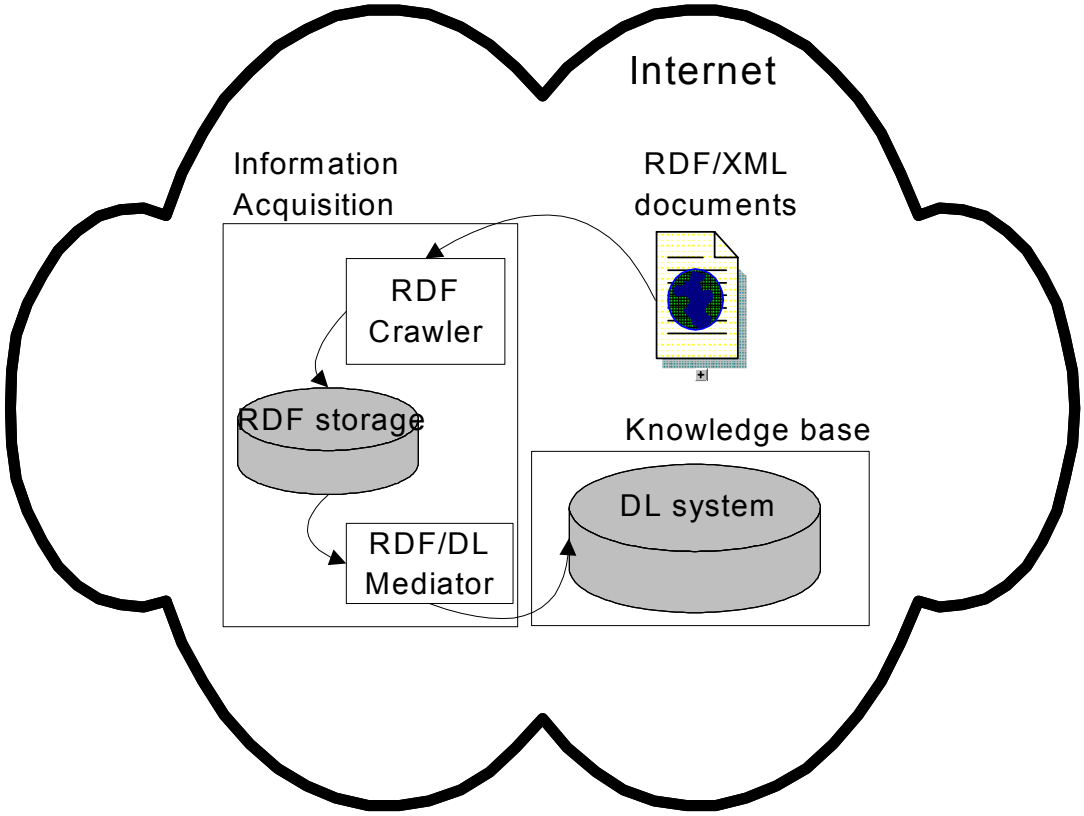


Figure 3.2. Information flow for information acquisition of the deduction system

3.2.1 Assumptions about the Data

This section summarizes the assumptions about the Semantic Web data that the deduction system will be based on. The assumptions are provided operationally when no standardized approach is available.

3.3.1.1 Resource Under the RDF Model and Syntax Specification [7], *resource* is defined as follows.

A resource may be:

1. *an entire Web page; such as the HTML document;*
2. *a part of a Web page; e.g. a specific HTML or XML element within the document source.*
3. *a whole collection of pages; e.g. an entire Web site.*

4. *an object that is not directly accessible via the Web; e.g. a printed book.*
Resources are always named by URIs plus optional anchor ids.

Given the definition, two kinds of resources may be distinguished: *retrievable resource* (definitions 1-3) and *non-retrievable resource* (definition 4). A retrievable resource is a resource whose content is accessible via the Web, e.g. HTML documents. A non-retrievable resource is a resource that is not directly accessible via the Web. This implementation accepts the distinction between two kinds of resources and will treat them differently in terms of how they may be identified and used. However, it does not elaborate the distinction in a greater detail.

3.2.1.2 Resource Identifier The URL has been an effective identifier scheme for resources on the Web. With the introduction of the Semantic Web paradigm, the exchange of information about resources is not only limited to retrievable resources but also non-retrievable resources. There is currently no standard for how the non-retrievable resources should be identified under the URI schemes.

There was an extensive discussion in the Semantic Web research community on how the non-retrievable resource should be identified [120]. Nevertheless, the agreement on the subject has not yet been reached. There have been two major viewpoints on the issue. One advocates the use of the URL scheme to identify non-retrievable resource. The other advocates the use of the URN scheme. The former approach has an advantage of reusing the existing identifier scheme on the Web, which could lead to an easier adoption. However, the fact that URL is designed for retrievable resources makes it unintuitive for non-retrievable resources. The latter approach has an advantage of providing name rather than location, which makes it more natural for non-retrievable resources. However, the lack of adoption of URN on the Web makes it a less attractive choice.

This implementation accepts the URN scheme as the means for identifying non-retrievable resources. The URN scheme is chosen because a URN represents a name rather than a location, which makes it more natural for identifying non-retrievable resources. Further, the resolution of

name to resource location will not be required for non-retrievable resources. In this implementation, the URN for non-retrievable resources is assumed to be in the form:

urn:X-<namespace_identifier>:<resource_identifier >

The *<namespace_identifier>* portion is assumed be provided in term of the domain name of the organization that is in charge of the namespace. For example, the URN "*urn:X-sis.pitt.edu:object1*" indicates that "*object1"* is the label name of an object which belong to the "*sis.pitt.edu"* namespace authority. The implementation makes no assumption about how the label name in each namespace should be designed. Nevertheless, it is assumed that each namespace must maintain the uniqueness of the identifier within the namespace.

3.2.1.3 Class, Property and Instance The operational definitions of *class*, *property* and *instance* are summarized as follows:

Class is defined as a type of resource in the RDFS specification. The resource representing a class must have an *rdf:type* property whose value is the resource *rdfs:Class*. A class identifier is assumed to be in a form of the location of the document where the class definition can be found plus the anchor id indicating the class name. For example, the identifier "http://foo.bar/x.rdf#ABC" identifies the class named "ABC" whose definition can be retrieved from the URL "http://foo.bar/x.rdf".

Property is defined as a type of resource in the RDFS specification. The resource representing a property must have an *rdf:type* property whose value is the resource *rdf:Property*. A property identifier is assumed to be in a form of the location of the document where the property definition can be found plus the anchor id indicating the property name. For example, the identifier "http://foo.bar/x.rdf#xyz" identifies the property named "xyz" whose definition can be retrieved from the URL "http://foo.bar/x.rdf".

Instances are the resources that are members of other classes, i.e. defined using *rdf:type*.

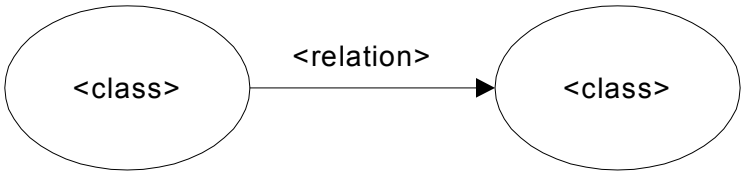
Instance identifiers are assumed to be differentiated between the instances that are retrievable and non-retrievable resources. In particular, the instance that is a retrievable resource is assumed to be identified by its networked location, i.e. a URL. The instance that is a non-retrievable resource is assumed to be identified by a name, i.e. a URN.

3.2.1.4 Relation and Attribute In RDF, a property can either be used to relate a resource to another resource or relate a resource to a literal. RDF does not make explicit distinction between the two kinds of property, i.e. both types are called *property*. The distinction is emphasized in this implementation.

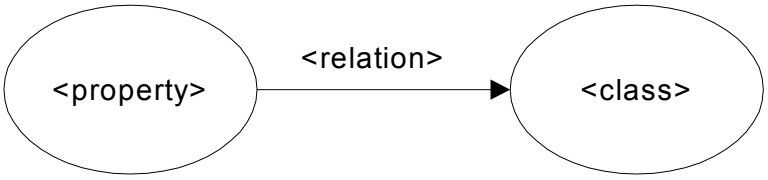
This implementation accepts the approach of distinguishing *relation* and *attribute* as two distinct kinds of RDF property [121]. The focus will be on *relation*, which describes association between two resources. An RDF property is considered a *relation* if it relates a resource to another resource in an RDF statement. An RDF property is considered an *attribute* if it relates a resource to a literal in an RDF statement.

3.2.1.5 RDF Statements RDF separates resource descriptions from vocabulary definitions. This has resulted in the separation of the RDF and RDFS specifications. However, the RDFS specification was defined using the RDF model and syntax. Thus, both forms share the same data structure that could be represented in terms of the RDF statements. The follows describe the processing rules in distinguishing the two kinds of RDF statements.

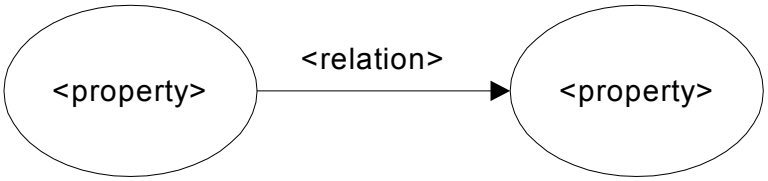
3.2.1.5.1 Vocabulary Definition Statement The implementation recognizes the forms of *vocabulary definition statement*, shown using the RDF graph notation, as follows:



e.g. *aaa rdfs:subClassOf bbb*, where *aaa* is a class and *bbb* is a class

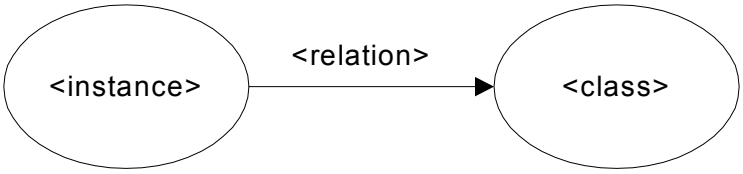


e.g. *ccc rdfs:domain aaa*, where *ccc* is a property and *aaa* is a class

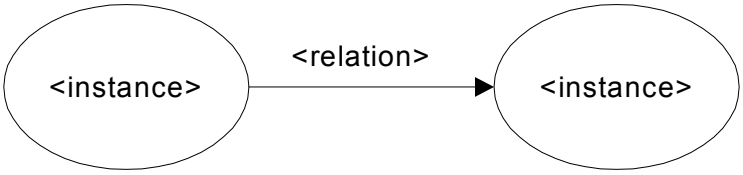


e.g. *ccc rdfs:subPropertyOf ddd*, where *ccc* is a property and *ddd* is a property.

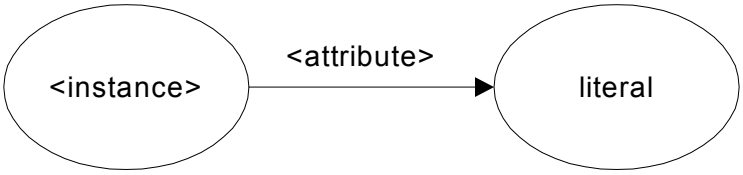
3.2.1.5.2 Resource Description Statement The implementation recognizes the forms of *resource description statement*, shown using the RDF graph notation, as follows:



e.g. *xxx rdf:type aaa*, where *xxx* is an instance and *aaa* is a class.



e.g. *xxx rdfs:seeAlso yyy*, where *xxx* is an instance and *yyy* is an instance.



e.g. *xxx rdfs:label xyz*, where xxx is an instance and xyz is a literal

It should be noted that, in RDF, a class can be defined an instance of another class. Similarly, a property can be defined an instance of a class. Such forms of RDF statement will be processed by the system as *vocabulary definition statement*.

3.2.1.6 Ontology Languages and Data Syntax The implementation supports the ontology data that is created using RDFS [28] and DAML+OIL [116]languages. It assumed that the RDF data uses the XML serialized syntax. An XML document representing the RDF data is refered to as an RDF/XML document. The implementation made the assumption that the RDF/XML documents are named with ".rdf" file extension to distinguish them from regular XML documents. Although the RDF data could also be embedded into the standard HTML documents, there is no standardized approach in embedding RDF data in HTML documents. The embedded RDF data in HTML is excluded from the processing of the system (see section 3.2.2 for further detail).

3.2.1.7 Data Decentralization An RDF/XML document could be placed on a HTTP server. Its location could be addressed by an URL. RDF/XML documents could be placed and retrieved on the Web, similar to HTML documents. RDF/XML document may be displayed on the Web browser as an XML data. Nevertheless, RDF/XML document is intended for automatic processing by computer programs rather than for human consumption.

RDF/XML document can be stored and retrieved in decentralized and modularized fashion. In particular, it can use the URL referencing system for redirecting computer programs to process additional sources of information. The implementation assumes the use of "*rdfs:seeAlso*" in providing the URL references in RDF/XML documents. For example, the following RDF/XML data indicates that additional information about the resource identified by the URN "urn:X-sis.pitt.edu:john_doe" could be retrieved from the RDF/XML document located by the URL http://foo.bar/sis_students_info.rdf.

```
<rdf:Description rdf:about="urn:X-sis.pitt.edu:john_doe" rdfs:label="John Doe">
    <rdfs:seeAlso rdf:resource="http://foo.bar/sis_students_info.rdf">
</rdf:Description>
```

3.2.2 RDF Crawler

Crawler or Web crawler is usually a program that gathers the information about the Web pages by traversing links. An RDF crawler was developed to acquire the information from the RDF/XML documents into the system. The RDF crawler only processes the RDF data in RDF/XML documents and ignores the RDF data embedded in HTML. The RDF information embedded in HTML is excluded due to several reasons. First, there is no standardized approach on how RDF data could be embedded in HTML. Further, by allowing the crawler to bypass the processing of HTML documents, the load of the crawler can be greatly reduced. The crawler recursively retrieves and processes every RDF/XML document referenced in the original RDF documents given a predefined traversal depth. The information collected by the crawler is stored in an RDF storage for further processing.

The development of the RDF crawler utilized the Jena toolkit[122] in the parsing of RDF/XML documents into the RDF statements. The RDF statements were stored in an RDF storage, which utilizes *mySQL3.2*[1] as its database backend.

3.2.3 RDF/DL Mediator

The RDF/DL mediator transformed RDF statements into description logic assertions. It also performed necessary encoding of resource identifiers in URI into the naming syntax allowed by the description logic system. It ignores any RDF statement that cannot be interpreted by the description logic system, i.e. statements containing literals. The processing rules involved in the transformation will be described in section 3.3.4.

[1] http://www.mysql.com/

3.3 KNOWLEDGE BASE

The declarative approach is used in constructing the system knowledge base. A declarative knowledge base contains a set of sentences in a formal language. Description logic was used as the formal language for the knowledge base.

There have been versions of description logic being developed. Each varies in the degree of the language expressiveness and computability performance of its inference procedure. $\mathcal{SHIQ}$ or $\mathcal{ALCQHI}_{\mathcal{R}+}$ is one of the expressive description logics based on the $\mathcal{ALC}$ language (see also section 2.2.7). The descriptions of $\mathcal{SHIQ}$ and its inference services are provided as follows.

3.3.1 $\mathcal{SHIQ}$

$\mathcal{SHIQ}$ is an extension of the $\mathcal{ALC}$ language, which is one of the most fundamental implementations of description logic. It extends the expressiveness of $\mathcal{ALC}$ with role transitivity ($_{\mathcal{R}+}$), qualified number restriction ($\mathcal{Q}$), hierarchy of role ($\mathcal{H}$) and inverse role ($\mathcal{I}$). As a result, $\mathcal{SHIQ}$ could also be called $\mathcal{ALCQHI}_{\mathcal{R}+}$. The following description of $\mathcal{SHIQ}$ is based on the descriptions provided in [123-125].

The formal language in $\mathcal{SHIQ}$ is composed of distinct sets of concept names ($\mathcal{CN}$), role names ($\mathcal{RN}$) and individual names ($\mathcal{O}$) together with a set of constructors for building concept expressions. The syntax and semantics of concept constructors in $\mathcal{SHIQ}$ is provided in Table 3.1. Concept is different from concept name in that a concept could either be a concept expression or a concept name. The syntax and semantics of roles in $\mathcal{SHIQ}$ are provided in Table 3.2.

The semantics of the DL syntax is given using the notion of Interpretation, which could be briefly introduced as follows. The interpretation $I = (\Delta^I , {}^I)$ consists of a non-empty domain (Δ^I) and an interpretation function (^{I}). The interpretation function could be applied to a concept, i.e.

$C^{I} = I(C)$, which maps a concept into a subset of Δ^{I}. The interpretation could be applied to a role, i.e. $R^{I} = I(R)$, which maps a role into a subset of the cartesian product of Δ^{I}, i.e. $(\Delta^{I} \times \Delta^{I})$. The interpretation function could be applied to an individual, i.e. $O^{I} = I(O)$, which maps an individual name into a member of Δ^{I}.

$\mathcal{SHIQ}$ concept expressions can be constructed using the combination of the following constructors: $\neg C$, $(C \sqcap D)$, $(C \sqcup D)$, $(\exists R.C)$, $(\forall R.C)$, $(\leq n\ R.C)$ and $(\geq n\ R.C)$, where C, D are concepts, R is a role, and n is an integer. Top $(\top)$ and Bottom $(\bot)$ are also concepts.

Table 3.1. Syntax and semantics of $\mathcal{SHIQ}$ concept constructors

Syntax	**Description**	**Semantics**
A	Concept name	$A^{I} \subseteq \Delta^{I}$
$\top$	Top	Δ^{I}
$\bot$	Bottom	$\varnothing$
$\neg C$	Negation	$\Delta^{I} \setminus C^{I}$
$C \sqcap D$	Conjunction	$C^{I} \cap D^{I}$
$C \sqcup D$	Disjunction	$C^{I} \cup D^{I}$
$\exists R.C$	Existential quantification	$\{x \mid \exists\, y\ (x, y) \in R^{I} \wedge y \in C^{I}\}$
$\forall R.C$	Universal quantification	$\{x \mid \forall\, y\ (x, y) \in R^{I} \Rightarrow y \in C^{I}\}$
$\leq n\ R.C$	Qualified number restriction	$\{x \mid \#\{\, y \mid (x, y) \in R^{I} \wedge y \in C^{I}\} \leq n\}$
$\geq n\ R.C$		$\{x \mid \#\{\, y \mid (x, y) \in R^{I} \wedge y \in C^{I}\} \geq n\}$

Note: # denotes the cardinality of a set

Table 3.2. Syntax and semantics of SHIQ roles

Syntax	Description	Semantics
R	Role name	$R^I \subseteq \Delta^I \times \Delta^I$
R^{-1}	Inverse role	$\{ (x, y) \in \Delta^I \times \Delta^I \mid (y, x) \in R^I \}$

The set of $\mathcal{RN}$ union the set of inverse roles (R^{-1}) is equal to the set of all roles in $\mathcal{SHIQ}$. Furthermore, two kinds of $\mathcal{RN}$ are distinguished: transitive role ($\mathcal{TRN}$) and functional role ($\mathcal{FRN}$). An $\mathcal{RN}$ is a $\mathcal{TRN}$ if it satisfies the following condition: for any $R \in \mathcal{TRN}$ if $(x, y) \in R^I$ and $(y, z) \in R^I$, then $(x, z) \in R^I$. An $\mathcal{RN}$ is a $\mathcal{FRN}$ if it satisfies the following condition: for any $F \in \mathcal{FRN}$ if $(x, y) \in F^I$ and $(x, z) \in F^I$, then $y = z$.

A $\mathcal{SHIQ}$ knowledge base ***K*** is a finite set of two kinds of statements: *terminological* and *assertional*. The set of the first kind of statements constitutes the ***TBox***. The set of the second kind constitutes the ***ABox***. This could be written as: ***K = { TBox ∪ ABox }***. The knowledge base may also be represented by the tuple ***K = < TBox, ABox>***.

The ***TBox*** contains the statements describing concepts and roles. The ***TBox*** statements are in the form shown in Table 3.3.

Table 3.3. Syntax and semantics of TBox statements

Syntax	Satisfied if
$C \doteq D$	$C^I = D^I$
$C \sqsubseteq D$	$C^I \subseteq D^I$
$R \sqsubseteq S$	$R^I \subseteq S^I$

where C, D are concepts, R, S are roles, The first form of the statements is used to indicate the equivalence between two concepts. The second form is used to indicate the subsumption relationship between two concepts. The third form is used to indicate the subsumption relationship between two roles. The formal meaning of the statements is given in terms of the

interpretation I = (Δ^I , ^{I}). An interpretation I satisfies the statement $C \doteq D$ if and only if $C^I = D^I$. An interpretation I satisfies $C \sqsubseteq D$ if and only if $C^I \subseteq D^I$. An interpretation I satisfies $R \sqsubseteq S$ if and only if $R^I \subseteq S^I$. [1]

The ***ABox*** contains the statements describing individuals. The ***ABox*** statements are in the form shown in Table 3.4.

Table 3.4. Syntax and semantics of ABox statements

Syntax	Satisfied if
a:C	$a^I \in C^I$
(a, b):R	$(a^I, b^I) \in R^I$

where C is a concept, R is a role, a and b are individual names. The first form of the statements indicates that an individual is an instance of a concept. The second form of the statements indicates that two individuals are related by a role. The formal meaning of the statements is given in terms of the interpretation I = (Δ^I , ^{I}). An interpretation I satisfies the statement a:C if and only if $a^I \in C^I$. An interpretation I satisfies (a, b):R if and only if $(a^I, b^I) \in R^I$.

3.3.2 Inference Services

The inference services in description logic are provided separately for the ***TBox*** and ***ABox***. This implementation requires the inference support for both the ***TBox*** and the ***ABox***, although the focus is more on that of the ***ABox***. The basic inference services for the ***TBox*** include subsumption, concept satisfiability and knowledge base satisfiability checking. The basic inference services for the ***ABox*** include instance checking, retrieval and realization. Table 3.5 provides a summary of inference services in description logic.

[1] $C \doteq D$ could also be expressed by $C \sqsubseteq D$ and $D \sqsubseteq C$.

Table 3.5. Summary of inference services in description logic

Inference service	Meaning	Example
Tbox		
Concept satisfiability	$K \models C \equiv \top \not\equiv \bot$	? Dog $\sqcap$ Animal
Subsumption	$K \models C \sqsubseteq D$	? Dog $\sqsubseteq$ Animal
Knowledge base satisfiability	$K \models$	? Dog $\doteq \neg$ Animal
ABox		
Instance checking	$K \models C(a)$	? Dog (snoopy)
Retrieval	$\{a \mid K \models C(a)\}$	Dog (snoopy) $\Rightarrow$ snoopy
Realization	$\{C \mid K \models C(a)\}$	Dog (snoopy) $\Rightarrow$ Dog

where $\models$ denotes *entailment*, aka logical implication

3.3.3 RACER

The implementation of the knowledge base utilizes the RACER (Renamed ABox and Concept Expression Reasoner) system [124]. The RACER system is a knowledge representation system that has a support for the description logic $\mathcal{ALCQHI}_{\mathcal{R}^+}$ or $\mathcal{SHIQ}$.

RACER was the first DL system that has reasoning support for both the ***TBox*** and the ***ABox*** for $\mathcal{SHIQ}$. All the standard inference services for ***TBox*** and ***ABox*** are supported by RACER. RACER also provides the set of commands that is compatible with the Knowledge Representation System Specification (KRSS) [126]. RACER has the unique name assumption for ***ABox***, i.e. two names can not refer to the same individual [127].

RACER has been implemented in Common Lisp. It provides the client-server interface for accessing the RACER system via the TCP/IP sockets. RACER also provides the RACER client interface in Java (JRacer) to allow an access to RACER system from Java applications. The implementation utilizes the RACER system version 1.7.6 running on a computer with Windows 2000 operating system, Pentium-733MHz and 1GB RAM.

3.3.4 RDF to DL Data Transformation

This section discusses the data transformation process made by the RDF/DL Mediator. The data transformation involves three mapping processes: RDFS to DL vocabulary mapping, DAML+OIL to DL vocabulary mapping and RDF statements to DL statements mapping.

3.3.4.1 RDFS to DL Vocabulary Mapping The RDFS vocabulary found in the acquired data was mapped into DAML+OIL vocabulary which was mapped into the description logic syntax subsequently. The mappings between RDFS and DAML+OIL vocabularies were straightforward as most of the primitive terms are defined as equivalences. The equivalences are listed as follows: *rdf:property = daml:property*, *rdfs:subClassOf = daml:subClassOf*, *rdfs:subPropertyOf = daml:subPropertyOf, rdfs:domain = daml:domain, rdfs:range = daml:range* and *rdf:type = daml:type*.

However, there was one exception for the *daml:Class*, which was defined in the DAML specification as a subclass of *rdfs:Class*. Practically, this implies that only classes that are defined in terms of *daml:Class* are equivalent to DL concepts. Those defined in terms of *rdfs:Class* will not always be equivalent to DL concepts. The implementation relaxed the constraint by operationally regarding them as equivalent, i.e. *rdfs:Class = daml:Class*. Although this does not fully satisfy the formal definition of *daml:Class*, the strictness was sacrificed in order to gain DL support for *rdfs:Class*.

3.3.4.2 DAML+OIL to DL Vocabulary Mapping The DAML+OIL vocabulary was transformed into DL concept and role constructors using the following mapping rules based on [128].

DAML+OIL *class* is equivalent to *concept* in DL. *Concept* in DL could be *concept name* or *concept expression*. Similarly, *class* in DAML+OIL could be *class name* or *class expression*. Class name is represented by a URI. Class expression could be formed using blank nodes and DAML+OIL class constructors. The mappings between DAML+OIL class constructors and concept constructors in DL are summarized in Table 3.6.

DAML+OIL *property*[1] is comparable to *role* in description logic. DAML+OIL provides a set of vocabulary for defining property. The mappings between DAML+OIL vocabulary and role constructors in DL are summarized in Table 3.7.

Table 3.6. Mappings between DAML+OIL vocabulary and DL concept constructors

DL	DAML+OIL vocabulary	DL Description
A	*<Class URI>*	Concept name
$\top$	*daml:Thing*	Top
$\bot$	*daml:Nothing*	Bottom
$\neg$C	*daml:complementOf*	Negation
C $\sqcap$ D	*daml:intersectionOf*	Conjunction
C $\sqcup$ D	*daml:unionOf*	Disjunction
$\exists$R.C	*daml:hasClass*	Existential quantification
$\forall$R.C	*daml:toClass*	Universal quantification
$\leq$ n R.C	*daml:maxCardinalityQ*	Qualified number restriction
$\geq$ n R.C	*daml:minCardinalityQ*	

Table 3.7. Mappings between DAML+OIL vocabulary and DL role constructors

DL	DAML+OIL vocabulary	DL Description
R	*<Property URI>*	Role name
R^{-1}	*daml:inverseOf*	Inverse role
TRN	*daml:transitiveProperty*	Transitive role
FRN	*daml:uniqueProperty*	Functional role

[1] Specifically, only the property that is considered *relation* is equivalent to *role* in DL. The property that is considered *attribute* is not.

3.3.4.3 RDF Statements to DL Statements Transformation The transformation between RDF statements and DL statements was performed straightforwardly. In particular, the RDF *vocabulary definition statements* (see section 3.2.1.5.1) were mapped into the ***TBox*** statements. The RDF *resource description statements* (see section 3.2.1.5.2) were mapped into the ***ABox*** statements. The RDF statements containing literals were ignored during the transformation.

The transformation of RDF statements into specific forms of ***TBox*** statements occurred through DAML+OIL vocabulary, which is summarized in Table 3.8.

Table 3.8. Mappings between RDF statements and DL TBox statements

DL	DAML+OIL vocabulary	DL Description
$C \doteq D$	*daml:sameClassAs*	Concept Equivalence
$C \sqsubseteq D$	*daml:subClassOf*	Concept Subsumption
$R \sqsubseteq S$	*daml:subPropertyOf*	Role Subsumption

The transformation of RDF statements into specific forms of ***ABox*** statements occured in two forms: instance-of statement and relationship statement. The first form is detected when the *rdf:type* is used as a property between an instance and a class. This form will be mapped to the ***ABox*** statement of *a:C*, i.e. *a* is an instance of concept *C*. The latter form is applied when an instance is related to another instance by a property. This form will be mapped to the ***ABox*** statement of *(a,b):R*, i.e. *a* and *b* are related by role *R*.

3.4 KNOWLEDGE RETRIEVAL

The Knowledge Retrieval component allows the retrieval of the gathered information and new conclusions from the knowledge base. It provides an application programming interface (API) to the system. The use of API allows the knowledge base to be accessible from various applications and environments. Further, the API allows the system to be independent of the underlying DL system. In particular, changes can be made to the underlying DL system and will be transparent to the application utilizing the API. In the current implementation, the Knowledge Retrieval

component could be considered a wrapper to the RACER system. It was built on top of the RACER client interface for Java (JRacer). Figure 3.3 provides a class diagram that summarizes the system API.

The API provides the retrieval interface via the two major classes: *ResourceClassifier* and *AssociationReasoner*. The *ResourceClassifier* class provides the query support for answering the questions related to classification of resources. The *AssociationReasoner* class provides the query support for answering the questions related to associations of resources. Each resource is an instance of the *Resource* class, which consists of a human-readable name and an identifier in the URI syntax. A human-readable name of a resource is obtained from the value defined using the “rdfs:label” attribute.

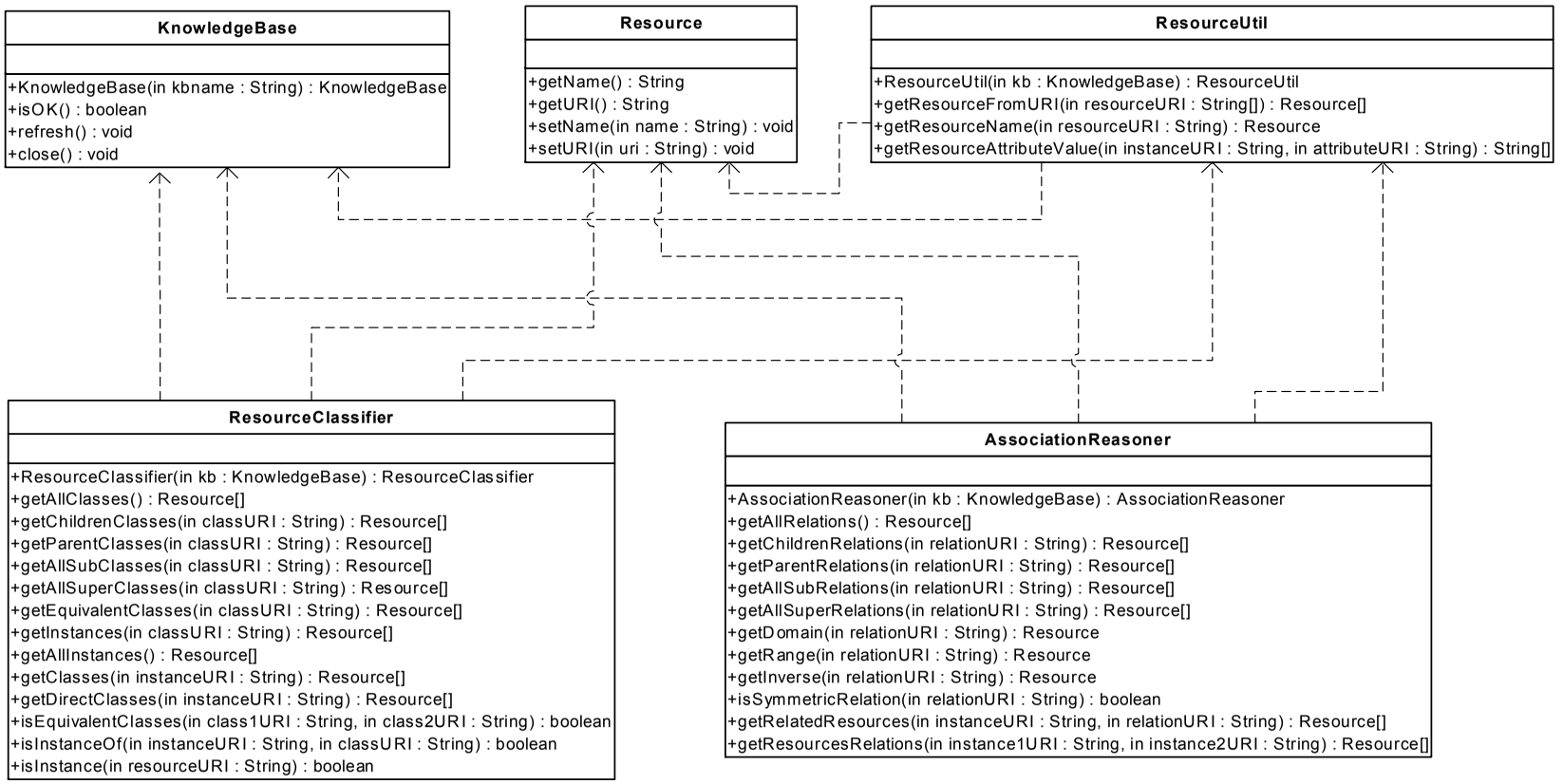


Figure 3.3. Class diagram for the Knowledge Retrieval component API

3.5 APPLICATION PROTOTYPE

This section provides a walk through an application prototype demonstrating the use of the deduction system for supplementing a Web resource collection. The course resources of the *INFSCI2770: Document Processing* in the School of Information Sciences at the University of Pittsburgh were used as the demonstrated resource collection. The description of the prototype is provided in terms of the three development processes: domain analysis, data creation and system deployment.

3.5.1 Domain Analysis

The domain analysis process involved the identification of classes, instances and relations. Classes were identified according to the characteristics of the course. Every entity of the course was considered to belong to the root class: COURSE-RESOURCE. Four major subclasses were subsequently defined: COURSE-ASSIGNMENT, COURSE-LECTURE, COURSE-DOCUMENT and COURSE-TOPIC. The class hierarchy is shown in Figure 3.4.

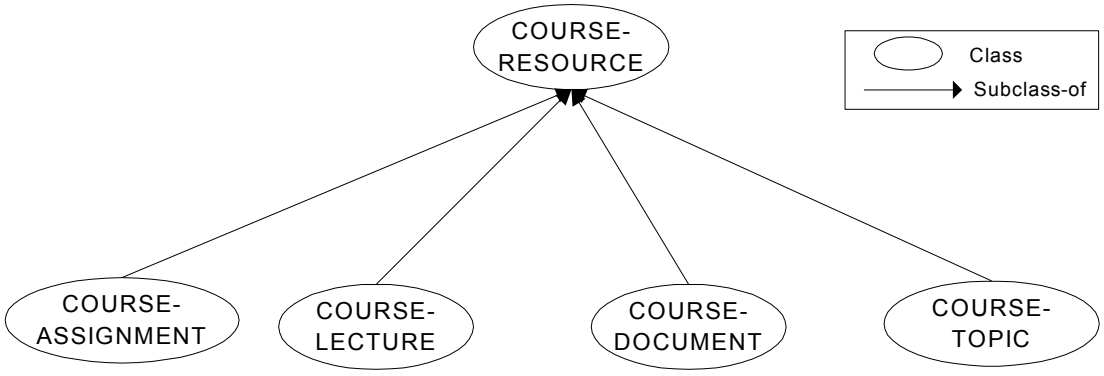


Figure 3.4. A class hierarchy for the course resources

The class hierarchy was further expanded from each leaf node. The COURSE-ASSIGNMENT and the COURSE-LECTURE nodes contain no subclasses. The COURSE-DOCUMENT node was further expanded into 15 subclasses shown in the Figure 3.5. The nodes with shading represent the defined classes, which are the classes that are defined in terms of other classes. The COURSE-TOPIC node was further expanded into 26 subclasses according to

the subject areas involved in the course. The class hierarchy for the course topics is shown in Figure 3.6.

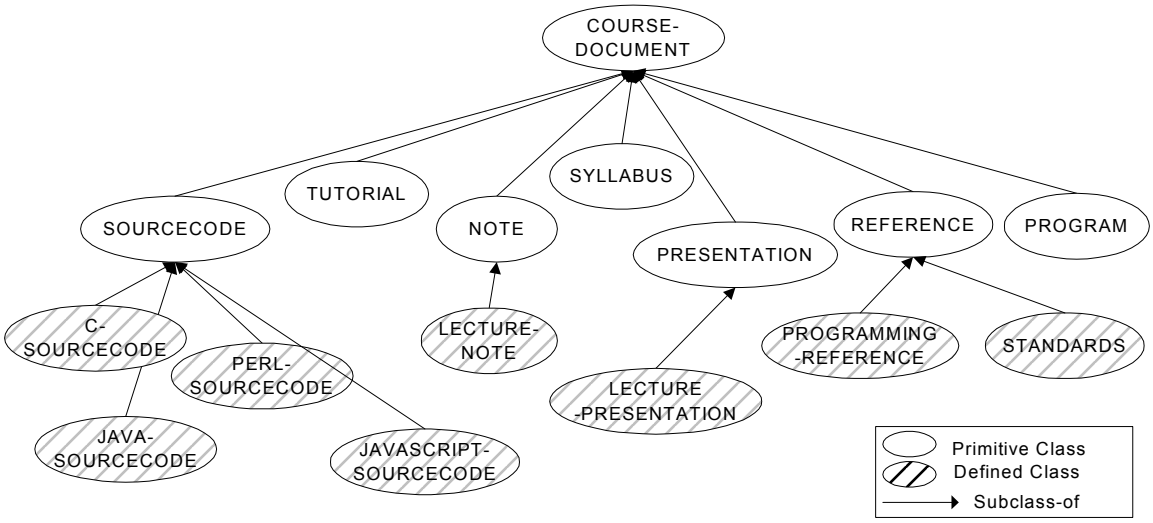


Figure 3.5. A class hierarchy for the course documents

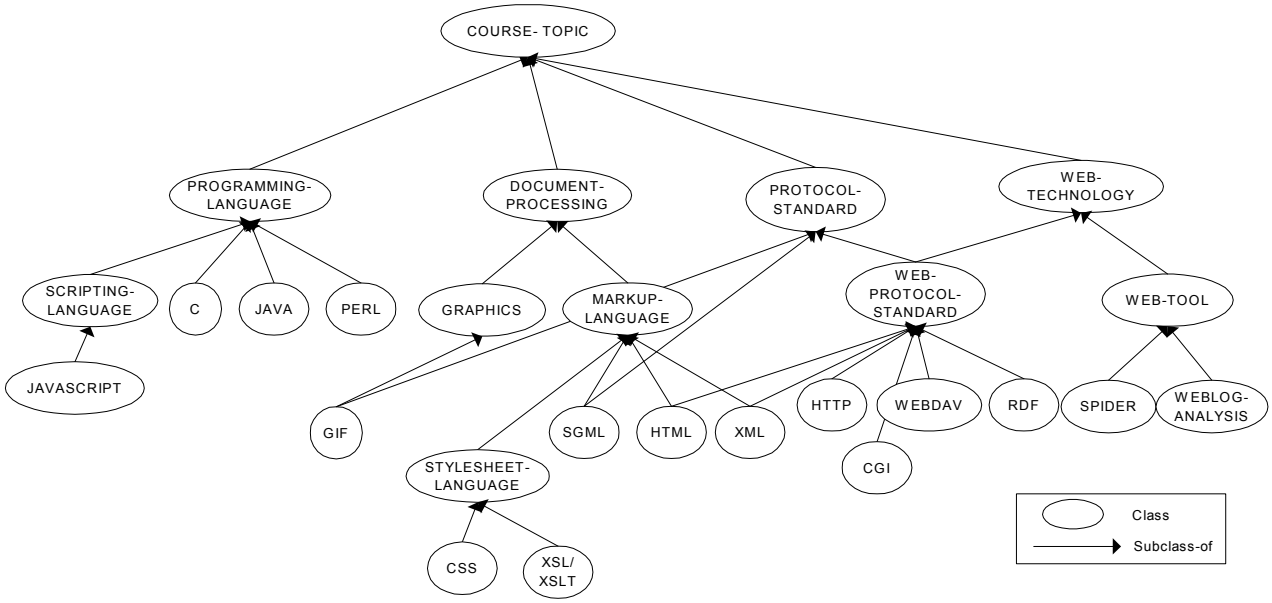


Figure 3.6. A class hierarchy for the course topics

Once the classes were defined, the next step involved the identification of the course instances. The six course assignments and the course final project were assigned as the instances of the COURSE-ASSIGNMENT class. The 14 course lectures were assigned as the instances of the COURSE-LECTURE class. The 16 course topics were assigned as the instances of the

COURSE-TOPIC. The 50 course documents were assigned as the instances of the COURSE-DOCUMENT class.

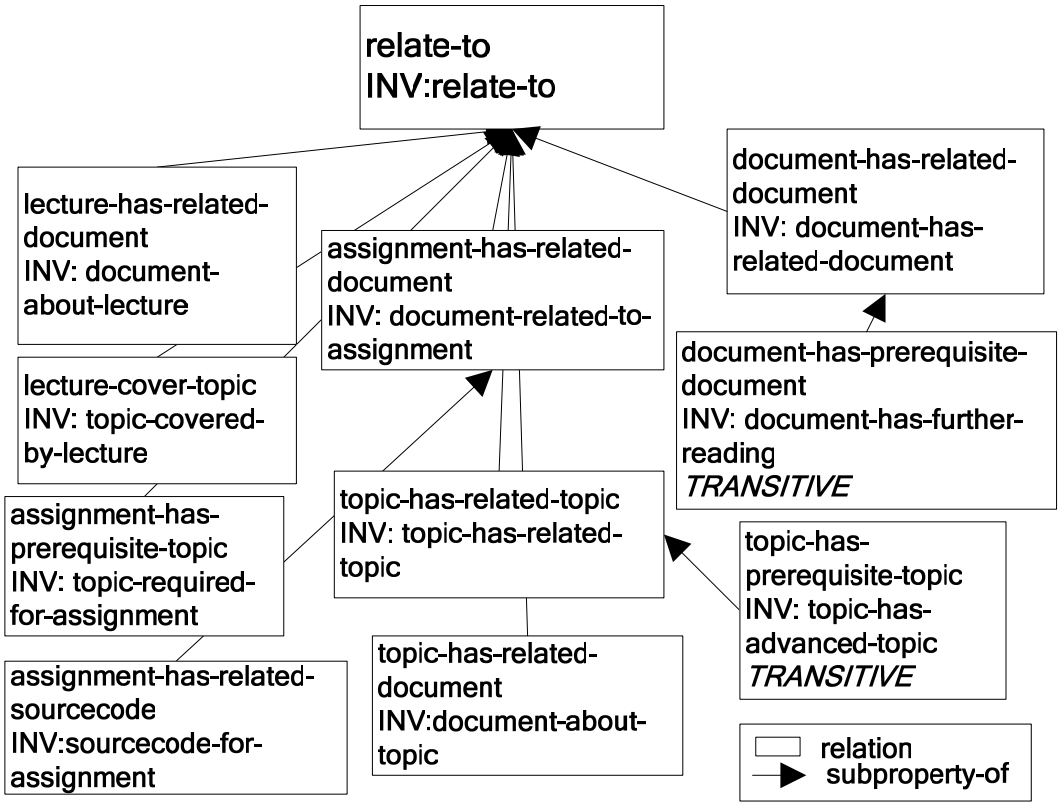


Figure 3.7. A relation hierarchy for the course resources

Relations were then defined such that the associations between the instances can be described. The main 11 relations were defined using the primitives of *subproperty-of*, *domain*, *range*, *inverse* and *transitivity*. The relations were defined and arranged into a relation hierarchy as shown in the Figure 3.7.

The total of 115 relationships between the course instances were indentified in terms of the modeled relations.

3.5.2 Data Creation

The results from the domain analysis process were captured into RDF/XML documents. Classes were expressed using *rdfs:Class*. Relations were expressed using *rdf:Property*. The subclass-of relationships between classes were expressed using *rdfs:SubClassOf*. The subproperty-of relationships were expressed using *rdfs:SubPropertyOf*. The instance-of relationships were expressed using *rdf:type*. The domain and range of a relation were expressed using *rdfs:domain* and *rdfs:range*. The inverse of a relation was expressed using *daml:inverseOf*. The transitive relations were expressed using *daml:TransitiveProperty*.

The names and the naming notation used during the domain analysis process were preserved, i.e. the class names used the upper-case letters, and the relation names used the lower-case letters. The resource identifier for an instance that is retrievable, e.g. a course document, is the document URL. The identifier for an instance that is non-retrievable, e.g. a course assignment, is the URN namespace plus the instance name written in lower-case letters with the first letter in upper-case letter. The URN namespace is simply defined as "*X-sis.pitt.edu*".

All the created data in RDF/XML documents were placed on a single Web server, "*http://talad.sis.pitt.edu:81"* in order to allow a convenient management of the documents. However, in reality, each document could also be independently located on different machines and referenced using URL (see section 3.2.1.7).

3.5.3 System Deployment

The RDF crawler was activated to collect the information from the created RDF/XML documents. The crawler traversed the referenced RDF/XML documents at two level depth of processing, which sufficiently covered the referencing depth in the created data.

The Web pages were created in order to allow the users to browse the information from the knowledge base using Web browsers. The Web pages were developed using the Java Server Page (JSP) technology. The JSP web pages used the Java API of the system to retrieve the

information from the knowledge base. The users can browse the collection of the course resources by the categories and by the relationships between them. The demonstration system was made accessible at the URL: *http://talad.sis.pitt.edu:81/demo/course_support/INFSCI2770*

3.5.4 Usage Scenarios

This section describes some usage scenarios of the demonstration system. The examples show some implicit information that was deliberately omitted and was discovered by the deduction system.

3.5.4.1 Sample use of deduction for the classification of information Figure 3.8 shows a sample scenario showing the use of deduction for the classification of the information resources. The "*Lecture Notes"* class has been defined as an equivalence of the "*Notes"* class that are "*related to*" some instances of the "*Lectures"* class. Three documents being described as the instances of the "*Notes"* class and are "*related to*" the lecture 1,2 and 3, which are instances of the "*Lectures*" class are classified into the category of "*Lecture Notes"* by means of deduction.

3.5.4.2 Sample use of deduction for the association of information Figure 3.9 shows a sample scenario showing the use of deduction for the association of the information resources. The "*Pre-requisite Readings"* relation was defined as a transitive relationship. Its inverse relationship was defined as the "*Advanced Readings"* relation. A document, "*spider1.c*", is described as a *pre-requisite reading* of the document, "*spider2.c*", which is subsequently described as a *pre-requisite reading* of the document, "*spider3.c*". Although not explicitly stated, when browsing the information of the document "*spider3.c*", the document "*spider1.c*" is also shown as one of its *pre-requisite readings*. The conclusion was made by the system based on the transitivity of the relationships. In addition, although not explicitly stated, the document "*spider3.c*" is also listed as one of the *advanced readings* of the document "*spider2.c*". The conclusion was made by the system based on the inverse of the relationships.

```
<!—DEFINITION OF THE DEFINED CLASS – LECTURE_NOTE-->
<rdfs:Class rdf:ID="LECTURE-NOTE" rdfs:label="Lecture Notes">
  <daml:intersectionOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#NOTE" />
      <daml:Restriction>
        <daml:onProperty rdf:resource="#document-about-lecture" />
        <daml:hasClass rdf:resource="#COURSE-LECTURE" />
      </daml:Restriction>
  </daml:intersectionOf>
</rdfs:Class>
```

```
<!—RESOURCE DESCRIPTION OF A NOTE -->
<rdf:Description                rdf:about="&docroot;Lectures/Lec_01-
_Introduction.txt"
      rdfs:label="Lecture note 01: Introduction">
   <rdf:type rdf:resource="&course;#NOTE" />
   <course:document-about-lecture
        rdf:resource="urn:X-sis.pitt.edu:is2770lecture1" />
</rdf:Description>
```

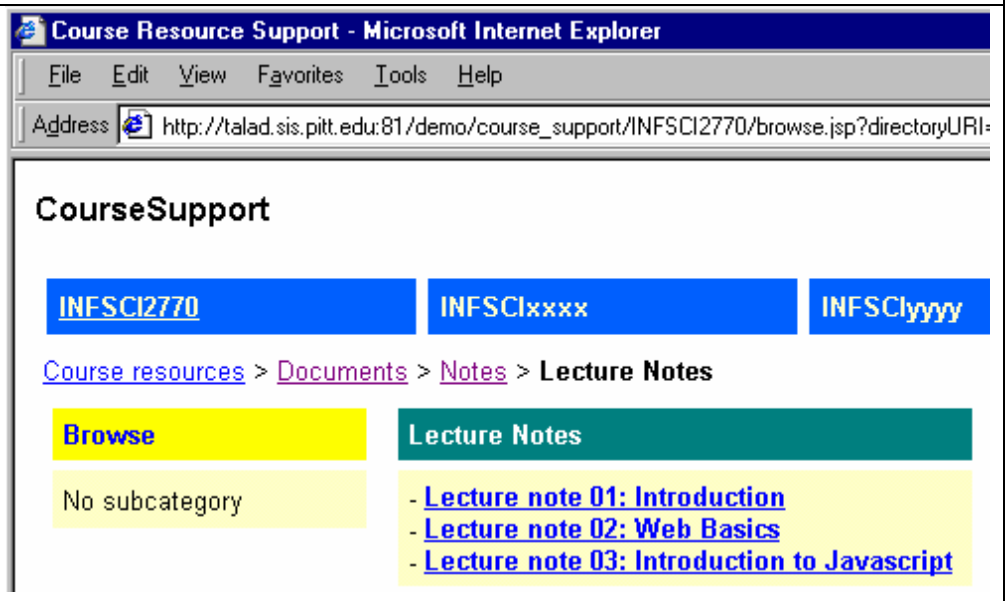


Figure 3.8. A use case for information classification by means of deduction

```
<!--PROPERTY DEFINITION -document-has-prerequisite-document-->
<daml:TransitiveProperty rdf:ID="document-has-prerequisite-document"
        rdfs:label="Pre-requisite Readings">
       <rdfs:domain rdf:resource="#COURSE-DOCUMENT" />
       <rdfs:range rdf:resource="#COURSE-DOCUMENT" />
       <daml:inverseOf rdf:resource="#document-has-further-reading" />
       <rdfs:subPropertyOf
                 rdf:resource="#document-has-related-document" />
</daml:TransitiveProperty>
```

```
<!--RESOURCE DESCRIPTION -SPIDER2.C and SPIDER3.C-->
<rdf:Description rdf:about="&docroot;Programs/C/Spider/spider2.c"
      rdfs:label="spider2.c">
             <course:document-has-prerequisite-document
     rdf:resource="&docroot;Programs/C/Spider/spider1.c" />
</rdf:Description>
<rdf:Description rdf:about="&docroot;Programs/C/Spider/spider3.c"
      rdfs:label="spider3.c">
              <course:document-has-prerequisite-document
     rdf:resource="&docroot;Programs/C/Spider/spider2.c" />
</rdf:Description>
```

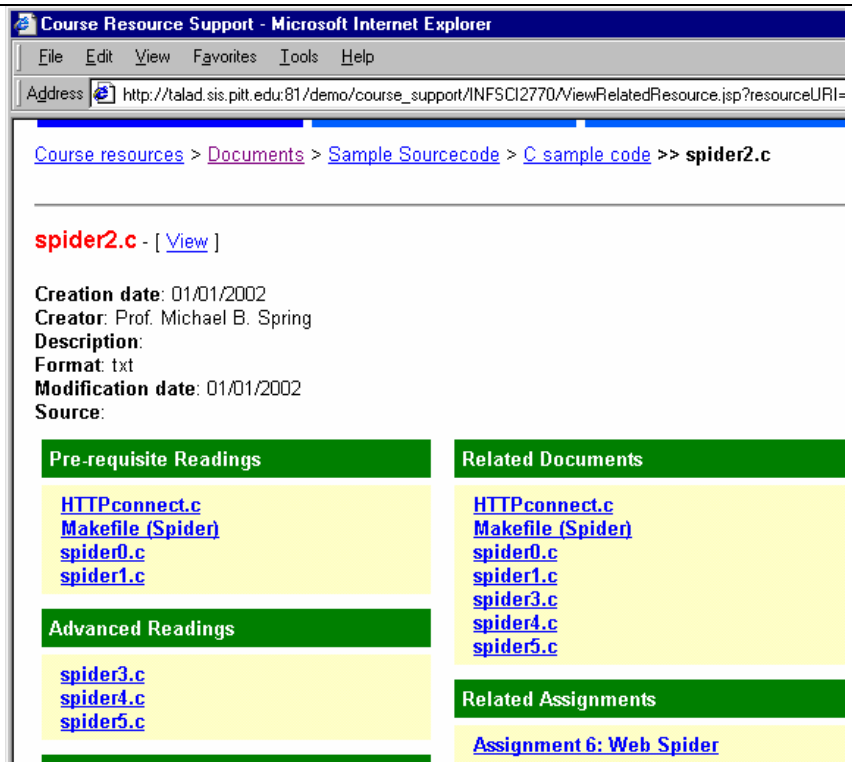


Figure 3.9. A use case for information association by means of deduction

3.6 CONCLUSION

In this chapter, the implementation of a deduction system for processing the Semantic Web data was presented. In addition, the development of an application prototype that verified the use of the system with a Web resource collection was presented. In the next chapter, the retrieval effectiveness of deduction system will be investigated using a case study conducted over a larger Web resource collection.

4.0 CASE STUDY

The case study provides a proof of concept design and assessment of a prototype for a resource collection employing deduction over classification and association. The information from a large book collection was used in providing the resource information for the deduction system. The effectiveness of the deduction techniques applied to the test collection was evaluated.

4.1 INTRODUCTION

A prototype developed in the early implementation suggested the potential of deduction applied over resource collections. However, the test collection was too small to allow the effectiveness of the deduction techniques to be assessed. Further, users had little difficulty in locating a resource in the small collection, thus the need to apply deduction was less emphasized. A larger collection of resources was needed to evaluate the impact of deduction system in the retrieval of information.

This case study investigated the effectiveness of the deduction techniques applied over a book collection, whose information is publicly available on the WWW. Although the study domain is limited to books, the techniques are applicable to all kinds of information resources that are similarly classified. The study produced a proof of concept system, evaluation results and a set of recommendations based on the evaluation results.

4.2 OBJECTIVES

The case study has the following objectives:

1) To demonstrate the use of resource descriptions and vocabulary definitions as a supplement to organize resource collections, whose information is publicly accessible over the WWW.
2) To evaluate the retrieval effectiveness of a deduction system.
3) To provide some guidelines, based on the results, for the implementation of a similar system in the context of the Semantic Web.

4.3 TEST COLLECTION

The book collection of the Amazon.com website was used as the test collection for the study. The Amazon.com book collection was chosen for the study for several reasons. First, the collection is "large" in size, offering a set of more than one million items. Second, the resources are classified and metadata are provided. Third, the information of the collection is publicly accessible over the World Wide Web. In particular, selected information can be accessed through the Amazon.com Web Service interface[1]. This interface was used for the acquisition of the information for the study.

It should be noted that the study uses the Amazon.com collection as a means to demonstrate and assess the uses of the deduction techniques over Web-based resource collections. It focuses on the deductive operations that occur in the decentralized fashion. Specifically, a deduction system acquires public information and operates on the information outside of the collection. Thus, although the collection at Amazon.com may be internally capable of processing similar operations, centralized processing is not the focus of this study. The deduction techniques used in

[1] Amazon.com Web Service (http://www.amazon.com/webservices/) is a platform that allows the retrieval of the information about Amazon.com product items in the structured data format over the WWW, i.e. through the XML over HTTP interface.

the study can be applied to resource collections regardless of their internal operations. The Amazon.com collection simply allows a simulation of these collections with controls.

4.4 DEDUCTION TECHNIQUES

The deduction techniques used in the study are classified into two major forms. The first form focuses on applying deduction over the classification system. In particular, the subject categories of the collection are used as the basis for deduction. The second form focuses on applying deduction over resource associations. In particular, the information on the resource properties is explored and used as the basis for deduction. The classes and relations for the resources in the collection are defined in the following sections.

4.4.1 Classes and Relations

The study defines six classes of resource entities for the resource collection: *Books*, *Topics*, *Publication Years*, *Authors*, *Publishers* and *Media Formats*. They are shown in Figure 4.1. The *Books* class consists of the book resources in the collection. Its classification system is based on the subject categories of the collection. The five other classes represent the entities related to the book resources. Although these classes are not the targets by themselves, they will be used to facilitate finding of book resources. The semantics of these classes will be additionally defined by the study and will be discussed in section 4.4.3.2.

Figure 4.1. Classes for the Amazon.com book resources

Resource properties provide specific information on the books such as title, author, publisher, publication date, media formats, and subject terms. Resource properties are used in the collection for the purposes of book identification similar to those of the traditional library catalog card. In the context of the study, resource properties are utilized as relations for relating the book resources with the resources of the defined classes. The relations are defined for the collection in terms of domains, ranges and inverse relations. The defined relations are shown in Figure 4.2.

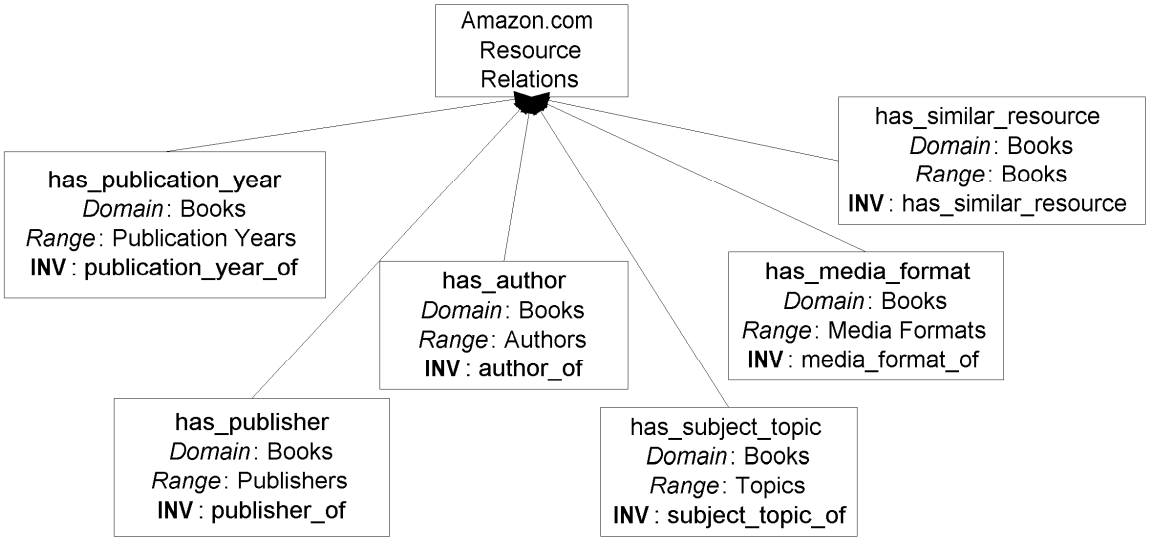


Figure 4.2. Relations for the Amazon.com book resources

4.4.2 Deduction over the Classification System

The books in the Amazon.com collection are classified by their subject categories. The classification system for the collection consists of 34 major subject categories[1]. The use of deduction techniques over the classification system may be described in the context of information retrieval. Specifically, information retrieval is a process where a user query is matched against a collection of resources and a subset is retrieved. Each subject category in the classification system could be considered a possible user query. When a user chooses a category, a set of resources is retrieved. For example, the *Science Fiction* category indicates a user query to find all the *science fiction books* in the collection. The listing of the book titles located under the *Science Fiction* category is considered the retrieved set of the resources that match the user query.

Retrieval in the classification system is most effective if the user's information need can be mapped directly to one of the existing categories. However, this is not always the case. If the information need does not match existing categories, the user usually has to choose the categories that are most relevant. For example, finding all the science fiction books published in the year 1994 cannot be represented in the form of an existing category in the collection. As a result, the user may need to choose the *Science Fiction* category and manually select those published in 1994. In such a case, the retrieved set of resources does not do a good job of meeting the user's information need.

Deduction could be applied when the information relevant to user's information need is not explicitly provided but is implicitly available in the classification system. Consider a user need to find all the *short-story science fiction books written by Arthur C. Clarke*. Although the *Short Story Fictions* category and the *Science Fiction Books by Arthur C. Clarke* category both exist in the classification system, neither provides the optimal response to the user need. Given the information in the classification system, the members of the abstract category *Short-story science fiction books written by Arthur C. Clarke* could be deduced based on the use of conjunction (Figure 4.3a). Similarly, the members of the abstract category *Non-short-story science fiction*

[1] Available at http://www.amazon.com/exec/obidos/tg/browse/-/1000 (accurate as of March 2004)

books written by Arthur C. Clarke could be deduced based on the use of negation (Figure 4.3d). The Figure 4.3 provides a summary and examples of the deduction techniques applied over the classification system.

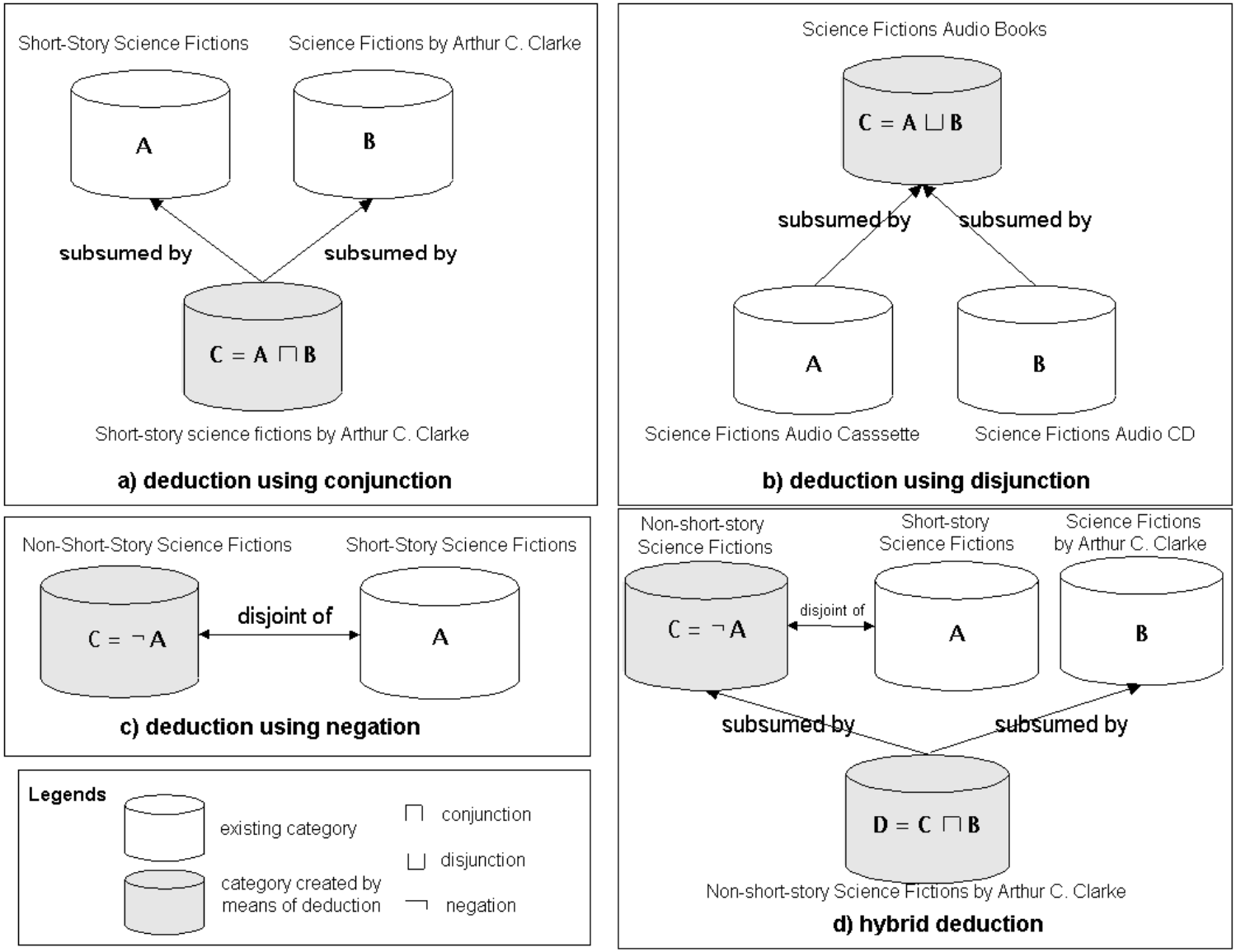


Figure 4.3 Deduction techniques applied over the classification system

4.4.3 Deduction over Associations

This section describes the deduction techniques applied over the associations between the book resources and the resources of the defined classes. It explores the information provided via the resource properties. To enable deduction over resource properties, the classes and relations of the resources were defined (see section 4.4.1). The information is the key that will allow additional inferences to be made about the resource relationships. The deductive operations applied to relations are described as follows.

4.4.3.1 Quantifier, Cardinality and Restrictions There are three operations which will be applied to relations: Quantifier, Cardinality and Restrictions. Quantifier could be used over a relation to indicate the existence of the relationships between the resources. Cardinality could be used over a relation to specify the number constraint of the relationships. Cardinality could be given in terms of minimum (at-least) or maximum (at-most) cardinality. Quantifier also implies cardinality. More specifically, the presence of quantifier implies a cardinality of at-least one. Examples of the uses of quantifier and cardinality are shown in Figure 4.4. In Figure 4.4a, an abstract category of "*co-authored books"* is defined as equivalent to the cardinality of at-least two over the *has_author* relation (≥ 2 *has_author*). Thus, a resource related to two or more authors is concluded a member of the abstract category of co-authored books.

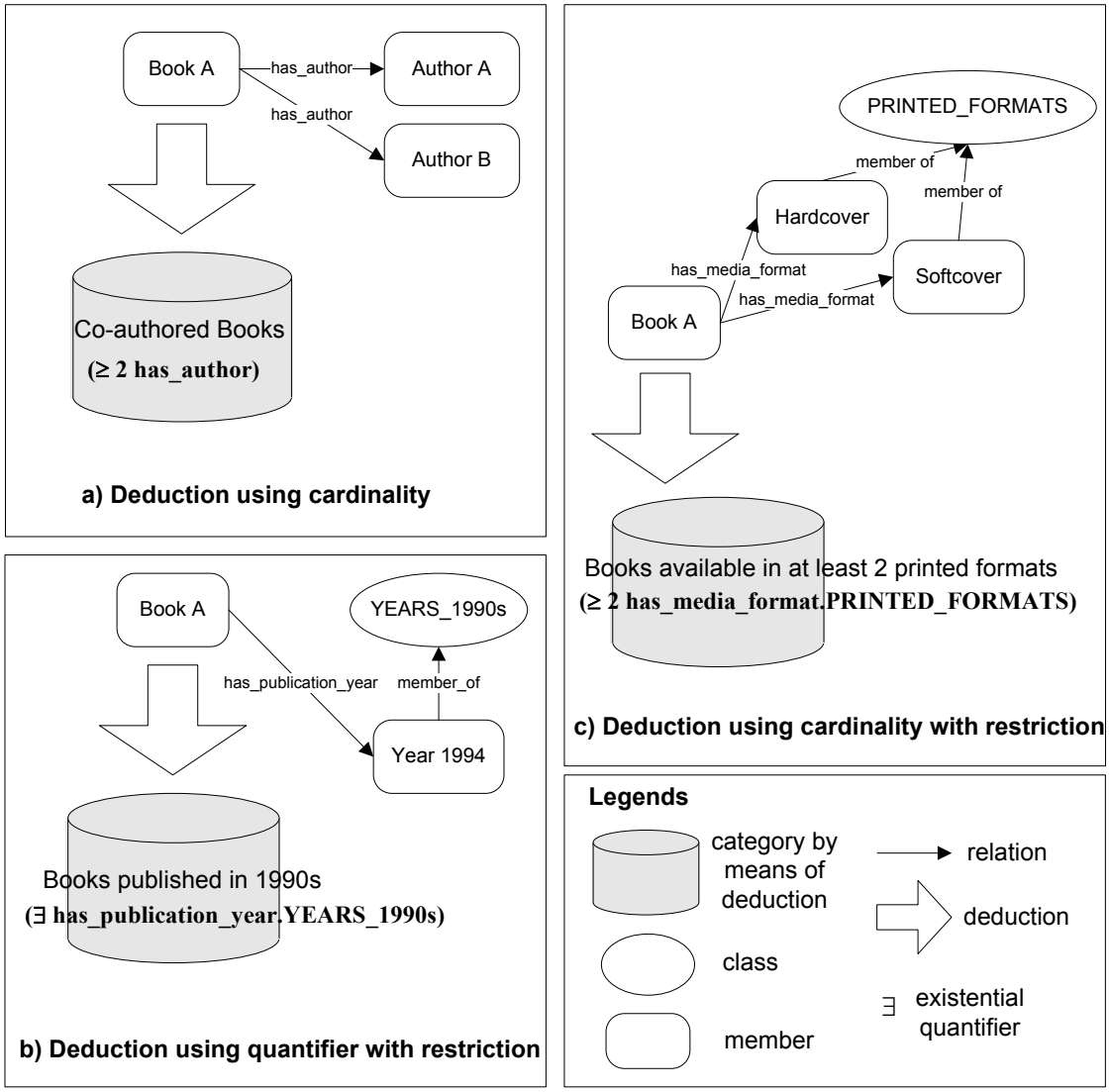


Figure 4.4. Deduction techniques applied over resource properties

In Description Logic, quantifier and cardinality are often used in combination with *restriction*. Restriction adds a constraint to the range of the relation. For example, in Figure 4.4b, an abstract category of *Books published in 1990s* is defined in terms of the quantifier over *has_publication_year* relation as well as its range restriction (*YEARS_1990s*). Thus, a resource published in the year 1994, defined as a member of the class *YEARS_1990s,* is concluded as a member of the category of books published in 1990s. Similarly, the use of cardinality with restriction is exemplified in Figure 4.4c. In particular, books available in more than one printed format could be represented using cardinality and restriction over *has_media_format* relation.

4.4.3.2 Deduction Based on Additional Semantics In Figure 4.4b and 4.4c, some added simple semantics were defined, i.e. the definitions of specific publication year and media formats. The added semantics are distinguished from the existing semantics in that they are created independent of the resource collection. Specifically, unlike the existing semantics, i.e. the subject categories, the creation of the added semantics is controlled by the study, not by the collection. These semantics are meant to simulate the inclusion of decentralized ontologies. The added semantics could vary from simple to complex forms.

4.4.3.2.1 Deduction Based on Added "Simple" Semantics In this study, simple semantics are defined for *Publication Years* and *Media Formats*. In the simplest form, the added semantics provide the classificatory semantics in terms of class members. In particular, the classificatory semantics of *Publication Years* is simply defined in terms of a set of individual year of a given time period. For example, the decade of the 1990s is simply defined in terms of each individual year of the decade. This allows a book published in the year 1994 to be concluded as a book published in the decade of the 1990s (Figure 4.4b).

Similarly but more elaborately, the classificatory semantics could be defined in hierarchical form. In particular, the semantics for *Media Formats* is defined in terms of a class hierarchy of the media formats available in the collection. For example, *Audiocassette* and *Audio CD* are two defined classes of media formats and are both subsumed by the defined class *Audio Book* format. In addition, the definitions of the disjointed classes are provided. For example, the defined class *Non-printed* format subsumes the *Audio Book* and the *Digital Book* media format classes. These allow the book titles available in the audiocassette format to be concluded as the book titles in audio book format as well as non-printed book format. The added classificatory semantics for the media formats are provided in Appendix A.1.

4.4.3.2.2 Deduction Based on Added "Complex" Semantics In a more complex form, the construction of semantics requires comprehensive domain knowledge. In the context of the collection, this is particularly true for those involved with the *Topics* of the books. The study

demonstrates the uses of added complex semantics by creating ones for the two subject topics: *Seafood Cookery*[1] and *US Presidents*.

With the added semantics defined for the topics, the subject categories of the collection could be further elaborated. In particular, based on the added *Seafood Cookery* semantics, the resources under the subject category "*Cooking / ByIngredient / Meat& Poultry& Seafood / Seafood*" could be further distinguished in terms of those related to *Fish Cookery* and *Shellfish Cookery*. For instance, the books in the category with the subject topic on salmon cookery could be concluded as those related to fish cookery. Similarly, the books with the subject topic lobsters cookery could be concluded as those related to shellfish cookery. Thus, the books with the subject topics related to fish cookery will be distinguished from those related to shellfish cookery. The added classificatory semantics for the *Seafood Cookery* subject topic are provided in Appendix A.2.

The added classificatory semantics for the *US Presidents* subject topic allows the books on individual US presidents under the subject category "*Biographies&Memoirs / Leaders&NotablePeople / Presidents&HeadsOfState*" to be recognized as those on the US president. Associative semantics which describe the relationships between each US president are also defined. In particular, the historical order of the US presidents in relation to each other and the existing biological relationships are defined. This allows the system to retrieve the resources on the subject. For example, the system can retrieve the book tiltles on the biography of the current US president based on book titles on the biography of the US president who has no successor defined. The added classificatory and associative semantics for the *US Presidents* subject topic are provided in Appendix A.3.

4.4.3.3 Deduction Based on Adhoc Associations The collection provides information about relationships between book titles called *similar resources*. The *has_similar_resource* relation was defined to represent such a relationship (Figure 4.2). The relation is defined as symmetric,

[1] The creation of the added semantics for the *Seafood Cookery* topic is based on the Library of Congress subject classification under the "*Cookery (Seafood)*" subject

i.e. if book *A* is associated with book *B*, book *B* is inferred to be associated with book *A*. In the context of the study, the associations established between the resources by such a relation are defined as "adhoc associations". Adhoc association is defined as an undefined relationship between the resources. If *A* is associated with *B* by an adhoc association, *A* may be semantically related to *B*, but the semantics are not defined[1]. Put another way, adhoc association provides minimal information for a semantic relationship.

This section explores deduction techniques applied over adhoc association. The goal is to use the adhoc associations to improve the retrieval effectiveness over an existing result set. In particular, the expanded results will be produced based on the degree of relatedness between the existing results and the adhoc resources associated with them. An expanded deduction result set (*D'*) is defined as:

$$D' = D \sqcup \geq n\ has\_similar\_resource.D$$

Where *D* is a result set obtained by various techniques. The expanded result set adds resources whose association degree with the original result set meets some criteria. Put more simply, *D'* includes the resources in the original result set *D* plus the resources in the collection which have an adhoc association with at least *n* resources in the result set *D*. In this study, the number of associated resources obtained for each resource is limited to five. Thus, *n* is an assigned integer between one and five.

4.5 PREPROCESSING FOR THE DEDUCTION SYSTEM

Preprocessing of the collection data was required to ensure that the information of the collection is delivered to the system in proper form. In particular, the information of the collection was

[1] In a sense, a hypertext link in the WWW created to convey a semantic relationship between two Web pages resembles an adhoc association. This is because the semantics of such a relationship can not be explicitly defined, and it may not exist.

provided given the closed-world condition. Further, the class and instance information was provided distinctly. Finally, valid identifiers are used.

4.5.1 Preprocessing Closed-world Information

This section addresses the preprocessing of the closed-world information for the knowledge base operating under the open-world assumption. The data from the Amazon.com collection is closed-world in nature. However, the deduction system is based on a description logic system, which operates under the open-world assumption. This has a significant impact on the results of the deduction system when negation, maximum cardinality and universal quantifier are involved.

4.5.1.1 Description Logic and the Open-world Assumption Description logic systems operate under the open-world assumption, i.e., no assumption is made about unknown information. In contrast, a system operating under the closed-world assumption assumes that any information unknown to the knowledge base is assumed to be false. A deduction system operating under the closed-world assumption must allow the conclusions in the knowledge base to be retracted once known to be false and therefore the deduction system is non-monotonic. With the open-world assumption, the knowledge base is monotonic because conclusions are only made based on the known information, thus retractions would not be needed. No description logic system deals with nonmonotonicity and thus none can operate under the closed-world assumption. This fact presents some problems for the study when negation and quantification are involved. The data was prepared as follows.

4.5.1.2 Preparing Data for Negation The results of a query involving negation depend on whether the open-world or the closed-world assumption is used. For example, consider the provided sets of members for the classes A, B and the deduction query expression: $(A \sqcap \neg B)$.

$A = \{a, b, c\}$

$B = \{a, b\}$

$(A \sqcap \neg B) = ?$

Using the closed-world assumption, *{c}* would be concluded to be the set of members for the abstract class *(A ⊓ ¬B)*. The deduction is made based on the assumption that the lack of the information that *c* is a member of *B* implies that *c* is not a member of *B*. Using the open-world assumption, it would be concluded that the members for the abstract class *(A ⊓ ¬B)* is an empty set. The deduction is made based on the assumption that even though *c* is not stated as a member of *B*, no assumption is made about *c* not being a member of *B*.

In the context of this study, the deduction result based on the closed-world assumption would be desirable. The example above presents the form of the information provided in the collection. The problem related to negation is that if the information is supplied to the deduction system directly, no result will be returned for the queries involving negation. In order to enable the deduction system to produce the results of those using the closed-world assumption, the explicit information related to the non-members of the class must be generated and provided to the deduction system. Using the example above, the information that *c* is not a member of *B* must be explicitly provided to the deduction system. The procedure in preparing this information may be exemplified as follows.

Using the example above, the universal set of the classes involved in the query may be created as:

$$U = A \cup B = \{a, b, c\}$$

Given the universal set, the complement of *B* may be obtained.

$$\overline{B} = \{x \in U | \, x \notin B\} = \{c\}$$

The information on the non-member of the class *B* can be additionally provided to the deduction system based on it.

$$\neg B = \{c\}$$

With the added information, the deduction system would produce the same result as those using the closed-world assumption:

$$(A \sqcap \neg B) = \{c\}$$

4.5.1.3 Preparing Data for Universal Quantifier and Maximum Cardinality Similarly, under the open-world assumption, the conclusions related to universal quantifier and maximum

cardinality usually cannot be made unless the maximum cardinality value is given. For example, consider the knowledge base containing the following two relation statements:

$a\ R\ b$

$a\ R\ c$

If the query expressions involving maximum cardinality and universal quantifier are made as follows:

$\leq 2\ R = ?$

$\forall R = ?$

A deduction system operating under the closed-world assumption would return $\{a\}$ as the results for both queries. This is based on the assumption that a does not relate to others other than those stated. Thus, the conclusion is that the maximum cardinality of a is equal to two via the relation R. In contrast, a deduction system operating under the open-world assumption would return the empty set for both queries. With the open-world assumption, no assumption is made related to the lack of information of other statements. In particular, with only the provided information, there is not enough evidence that a does not relate to some other resources. Thus, no conclusion could be made related to the maximum cardinality of the resource.

In the context of this study, the deduction result based on the closed-world assumption would be desirable. The example above presents the form of the information available in the collection. Thus, if the information is supplied directly, the deduction system will not produce the expected results when maximum cardinality and universal quantifier is involved. In order to enable the deduction system to produce the desirable outcomes, the information related to the cardinality of the resources must be generated and explicitly provided to the system. The procedure in preparing such information could be exemplified as follows.

Given the above example and the closed-world assumption, it is true that:

$a \in \{x \mid \#\{y \mid (x, y) \in R\} = 2\}$, where # denotes the cardinality of a set

This implies that a is a member of the class $\geq 2R$ as well as a member of the class $\leq 2R$. The information could be created and supplied to the deduction system.

With the added information, the deduction system would produce the same results as those using the closed-world assumption:

$\leq 2\ R = \{a\}$

$\forall R = \{a\}$

4.5.2 Preprocessing Class and Instance Information

In description logic, a clear distinction is made between class and instance. In this study, the resource property values in the collection are represented to the deduction system in terms of instances. For example, the information that a book title is in *Hardcover* media format is represented to the deduction system in terms of two related instances. However, in providing the semantics for the media formats, *Hardcover* is defined as a class being subsumed by the *Printed Format* class (Figure A.1). In such a case, *Hardcover* is represented to the deduction system as a class. Put another way, *Hardcover* needs to be represented to the deduction system as a class as well as an instance in different circumstances. In order to enable such uses, the study defines the entity as a class consisting of an instance representing the same entity as its only member. For example, in the above scenario, *Hardcover* media format is defined as a class with the instance *Hardcover* media format as its only member.

Similarly, in providing the semantics for the *US President* subject topic, each entity must be represented as a class as well as an instance in different circumstances. In particular, in providing the classificatory semantics, each US president entity must be represented in terms of a class (Figure A.3). However, in providing the associative semantics, each US president entity must be represented in terms of an instance (Figure A.4). Thus, the study defines each entity as a class consisting of an instance representing the same entity as its only member. For example, the entity representing the former US president, Ronald Reagan, is defined as a class denoted by *RONALD_REAGAN* as well as an instance denoted by *Ronald_Reagan* as its only member. Thus, given a concept expression, *($\exists$ is-predecessor-of.RONALD_REAGAN*), the list of the US president instances, which relate to the instance *Ronald_Reagan* over the *is-predecessor-of* relation, could be returned accordingly.

4.5.3 Preprocessing Identifier Information

In this study, the identifiers for the book titles provided to the deduction system were based on the resource identifiers used by the collection. The Amazon.com collection uses the Amazon.com Standard Item Number (ASIN) as the unique identifier for its resources. An ASIN is ten characters long and may consist of letters or digits. The identifier of a book title was provided to the deduction system as an ASIN string plus the "*i*" prefix[1]. The identifiers for the subject categories used by the deduction system were also provided based on those used by the collection. In particular, a subject category identifier was provided as the category identifier used by the collection, which is a unique number, plus the "*C*" prefix.

The property values of the resources were presented to the deduction system in terms of instance identifiers. However, achieving this was not straightforward because the property values provided by the collection are in the form of literal strings. For example, a book has an author named "Arthur C. Clarke". However, the author name could not be supplied directly to the deduction system and must be converted into the instance identifier in the proper form. In particular, the preprocessing replaced the exception characters, e.g. whitespace character, with the character "_" and the "*i*" prefix was added to the encoded name.

In some rare cases, the resource property values provided by the collection for the same entity were inconsistent. For example, although the term "shrimp" is often used in the subject terms, the term "shrimps" is also found in a few cases. The inconsistency was also found in an author name, i.e. "Arthur C. Clarke", "Clarke, Arthur C.", "Arthur Charles Clarke", "Arthur Clarke". In order to control such inconsistency to affect the performance of the deduction system, the study created a dictionary to transform the spelling variations of some terms into unique identifiers. For example, the variations of name used for Arthur C. Clarke in the collection will be transformed into the unique identifier "*iArthur_C_Clarke*" before supplying it to the deduction system.

[1] The prefix is necessary because the RACER system does not accept instance name beginning with digits

4.6 ANALYSIS OF THE RETRIEVAL EFFECTIVENESS OF THE DEDUCTION SYSTEM

This section describes the experimental design for assessing the retrieval effectiveness of the deduction system. The major research question is whether the use of the deduction techniques will result in more effective retrieval in the test collection.

4.6.1 Definitions

4.6.1.1 Deduction Query, Control Set and Deduction Result Set A *deduction query* is a request to the deduction system to retrieve a set of resources according to some specific information need. In this study, the deduction query is represented in the form of concept expression in description logic syntax.

The *control set* is a set of resources in the collection that the deduction system operates upon to obtain the results to a query. Specifically, it defines the target subject categories for the deduction system to operate upon. The control set may be different for each query.

The deduction system acquires information about the resources in the control set. It produces a *deduction result set* consisting of the resources that match the deduction query expression.

4.6.1.2 Retrieval Effectiveness The retrieval effectiveness of the deduction system will be assessed in terms of the *precision* and *recall* of the result set. The precision and recall of a result set are defined as:

$$Precision = \frac{\text{Number of relevant resources in the result set}}{\text{Number of resources in the result set}}$$

$$Recall = \frac{\text{Number of relevant resources in the result set}}{\text{Number of relevant resources in the control set}}$$

The accuracy of the deduction system in retrieving the resources is measured in terms of precision. Precision is the proportion of the resources in the result set that are relevant. In addition, the effectiveness of the deduction system in retrieving the relevant resources from the control set will be measured. Recall is the proportion of the relevant resources in the control set that are retrieved by the deduction system.

4.6.2 Hypotheses

The study will assess the effectiveness of the deduction system in retrieving the resources in the test collection. The assessment will be conducted based on a number of queries defined by the study. The precision and recall of the deduction result set against each query will be measured. The hypotheses for the assessment are stated as follows:

The first null hypothesis:
H0) The precision of the deduction result set against the query is equal to one
The alternate hypothesis:
H1) The precision of the deduction result set against the query is less than one.

The second null hypothesis:
H0) The recall of the deduction result set against the query is equal to one.
The alternate hypothesis:
H1) The recall of the deduction result set against the query is less than one.

4.6.3 Methodology

4.6.3.1 Procedure A set of queries was defined in terms of the information needs related to particular subjects. The queries were defined such that the query results could not be directly obtained from the collection but could be obtained based on some deduction techniques. The queries were mapped into the concept expressions in description logic syntax. The control set was also defined for each query. The control set was the union of the members of the subject

categories used in the query expression. The deduction system acquired the necessary information about the resources in the control set from the collection. The queries were run against the deduction system for instance retrieval of the concept expressions. The result sets were assessed for their relevancy to the queries. The retrieval effectiveness, in terms of precision and recall, was measured accordingly.

4.6.3.2 Queries Seventy five queries were defined. The query descriptions as well as the query expressions are listed in Appendix B.1. The classification of the queries is summarized in Tables 4.1-4.3.

Table 4.1. Classification of the queries by deduction techniques

Deduction techniques	**Number of queries**
1) Deduction over the classification system (section 4.4.2)	32
2) Deduction over resource properties without added semantics (section 4.4.3.1)	8
3) Deduction over resource properties with added simple semantics (section 4.4.3.2.1)	7
4) Deduction over resource properties with added complex semantics (section 4.4.3.2.2)	16
5) Deduction based on adhoc associations (section 4.4.3.3)	12

Table 4.2. Classification of the queries by query expressiveness

Query Expressiveness[1]	Number of queries
1) Conjunction	63
2) Disjunction	31
3) Negation	15
4) Quantifier	27
5) Cardinality	16

Table 4.3. Classification of the queries by subject areas

Subject areas	Number of queries
1) Computers & Internet	9
2) Biographies & Memoirs	16
3) Art History & History	8
4) Cooking	20
5) Travel	7
6) Science Fiction & Fantasy	9
7) Literature & Fiction	6

4.6.3.3 Assessing the Results Returned To measure the precision of a result set, the total number of the relevant resources in the result set must be obtained. To measure the recall of a result set, the total number of the relevant resources in the control set must be additionally obtained. Ideally, measuring the number of the relevant resources in any set is accomplished by reviewing and assessing each member of the set. Because such exhaustive examination would prove excessively costly for large result sets, a methodology was developed to estimate the number of relevant resources in large result sets. The proposed methodology is based on statistical inferences where the estimations of the actual measures are made based on the

[1] The query expression is in the normalized form, e.g. $\neg (A \sqcap B)$ was normalized to $\neg A \sqcup \neg B$, before being examined.

observations of the random samplings. This section describes the various techniques used and the problems associated.

4.6.3.3.1 Sampling Method Statistically, the sample size required to reliably predict the number of relevant resources in a result set is dependent upon the expected frequency of occurrences of the relevant resources in the result set. Put more simply, a larger sample size is required when the expected frequency of occurrences is smaller. In the context of this study, the sample size (n) is considered sufficiently large when $np_r \geq 5$ and $nq_r \geq 5$ [129] (p.13), where p_r is the expected proportion of the relevant resources in the result set and $q_r = 1 - p_r$. When the sample size is sufficiently large, the confidence interval of the estimation is more accurate.

For example, to reliably predict the number of the relevant resources in a result set where about 50% of the resources are expected to be relevant ($p_r = 0.5$), the minimum sample size of 10 is recommended. When only 5% of the resources are expected to be relevant ($p_r = 0.05$), the minimum sample size of 100 is recommended. The Table 4.4 shows some minimum sample sizes recommended for various p_r.

Table 4.4. Recommended minimum sample sizes for some expected proportion of relevant resources

Expected proportion of the relevant resources (p_r) [1]	**Minimum sample size (n)**
0.5	10
0.2	25
0.1	50
0.05	100
0.02	250
0.01	500
0.005	1000

[1] The suggested sample size is reversed when the predicted proportion is more than 50%. For example, when 95% of the resources are expected to be relevant ($q_r = 0.05$), the minimum sample size recommended is 100.

In this study, a fixed sample size of 500 will be used in for the estimation in the large sets. This sample size is chosen for two reasons. First, the sample size of 500 allows the confidence interval of an estimation to be reliably created when 1-99% of the resources in the set are expected to be relevant. In the context of the study, the ratio between the deduction result set size and the control set size is used as a rough prediction of the expected proportion of the relevant resources in the control set. Based on the preliminary results, the obtained ratios suggest the predicted proportion to be well within the range. Thus, the sample size of 500 is assumed sufficiently large for the large control sets. Although no prediction is made priori about the expected proportion of the relevant resources in the deduction result set, it is expected to be high. Thus, the sample size of 500 is also assumed sufficiently large for the large result sets. Second, using a fixed sample size allows the confidence interval to be created more consistently for different sets in comparison to the varied sample sizes.

In summary, when the deduction result set contained 500 items or less, all the items in the set were reviewed and assessed for their relevancy. When the deduction result set contained more than 500 items, 500 items were randomly chosen for the review. The same procedure was also applied to the control sets. In particular, when the control set contained 500 items or less, all the items in the set were reviewed and assessed for their relevancy. When the control set contained more than 500 items, 500 items were randomly chosen for the review.

4.6.3.3.2 Relevance Judgment The resource relevance judgment was made by a panel consisting of three external judges. The rationale was that a resource that was judged as relevant by at least two judges would be considered a relevant resource. Given this requirement, the resources were initially assessed by two judges. If the two judges agreed on the relevance of a resource, the resource was considered a relevant resource and no further assessment was required. When the two judges disagreed about the relevance of a resource, the resource was further reviewed and assessed by the third judge. The resources assessed as relevant by the third judge was considered relevant.

The details on the recruitment of the judges for the study are provided in Appendix E.1.

4.6.3.3.3 Retrieval Effectiveness Assessment

4.6.3.3.3.1 Precision When all the resources in a deduction result set were reviewed and assessed, the proportion of the relevant resources found in the set defines the precision of the result set. When only samples of a deduction result set were reviewed and assessed, the proportion of the relevant resources found in the sample set is used to estimate the precision of the result set. In this case, the estimation is used along with the 95% confidence interval of the estimation. The lower and upper limits of the confidence interval are measured using the following formulas [130]:

Lower limit: $$L = \frac{2np + z_{\alpha/2}^2 - 1 - z_{\alpha/2}\sqrt{z_{\alpha/2}^2 - 2 - 1/n + 4p(nq+1)}}{2(n + z_{\alpha/2}^2)}$$

Upper limit: $$U = \frac{2np + z_{\alpha/2}^2 + 1 + z_{\alpha/2}\sqrt{z_{\alpha/2}^2 + 2 - 1/n + 4p(nq-1)}}{2(n + z_{\alpha/2}^2)}$$

Where p is the proportion of the relevant resources found in the sample set, n is the sample size (=500), q = 1-p, $\alpha = 0.05$ and $Z_{\alpha/2} = 1.96$. However, if $p = 0$, L is set to 0 and if $p = 1$, U is set to 1.

4.6.3.3.3.2 Recall When all the resources in a control set were reviewed and assessed, the proportion of the total relevant resources in the deduction result set compared to the total relevant resources in the control set provides the recall of the result set. However, when the control set is large and the samples must be used, estimating the recall of the result set is not straightforward. In this study, two methods of estimating the recall of a result set were used. The final estimated recall was the average on the estimated recall obtained from both methods. The two methods of estimating a recall are described as follows.

Method 1 When the actual number of the total relevant resources in the deduction result set can be obtained, the estimated recall is defined as:

$$\text{Estimated Recall} = \frac{\text{Number of relevant resources retrieved}}{\text{Estimated number of relevant resources}}$$

$$\text{Lower Limits: } \frac{\text{Number of relevant resources retrieved}}{\text{Upper limit on the estimated number of relevant resources}}$$

$$\text{Upper Limits: } \frac{\text{Number of relevant resources retrieved}}{\text{Lower limit on the estimated number of relevant resources}}$$

The estimated number of relevant resources is equal to $p_c N_c$, where p_c is the proportion of relevant resources found in the samples of the control set and N_c is the control set size. The lower and upper limits of the confidence interval are $L_c N_c$ and $U_c N_c$ accordingly, where L_c and U_c are the lower and upper limits of the confidence interval for p_c which can be obtained from the formulas defined in section 4.6.3.3.3.1.

However, when the deduction result set is large and the total relevant resources in the result set must be estimated, the estimated recall is defined as:

$$\text{Estimated Recall} = \frac{\text{Estimated number of relevant resources retrieved}}{\text{Estimated number of relevant resources}}$$

$$\text{Lower Limits: } \frac{\text{Lower limit on the estimated number of relevant resources retrieved}}{\text{Upper limit on the estimated number of relevant resources}}$$

$$\text{Upper Limits: } \frac{\text{Upper limit on the estimated number of relevant resources retrieved}}{\text{Lower limit on the estimated number of relevant resources}}$$

The estimated number of relevant resources retrieved is equal to $p_d N_d$, where p_d is the proportion of relevant resources found in the samples of the result set and N_d is the result set size. The lower and upper limits of the confidence interval are $L_d N_d$ and $U_d N_d$ accordingly, where L_d and U_d are the lower and upper limits of the confidence interval for p_d which can be obtained from the formulas defined in section 4.6.3.3.3.1.

Method 2 The second method of estimating recall uses negative evidence in predicting recall. In particular, it additionally uses the number of relevant resources that were not retrieved by the deduction system to measure the recall. In this method, Recall is defined as:

$$\text{Recall} = \frac{\text{Total relevant resources retrieved}}{\text{Total relevant resources retrieved} + \text{Total relevant resources not retrieved}}$$

When the control and result sets are small and no estimation is required, the recall measured using this method will be identical to that measured using the first method. However, when samples of the sets are used, both methods could give different estimation on the recall. The difference depends on the proportion of the relevant resources in the sample sets that were found not to be retrieved by the deduction system.

Consider a case when the system retrieves five resources and all of them are found relevant. To measure recall, five relevant resources are found among the sample of the control set. Using the first method, the evidence would lead to the estimation that the recall could be one. However, using this method, one relevant resource in the sample is found non-retrieved by the system. Based on the new evidence, the recall must be less than one, in proportion to the number of relevant resources found not to be retrieved.

In this method, the number of relevant resources is the sum of the number of relevant resources retrieved and the number of relevant resources not retrieved. Thus, when the result set is small but the control set is large, the estimated recall could be defined as:

$$\text{Estimated Recall} = \frac{\text{Total relevant resources retrieved}}{\text{Total relevant resources retrieved} + p_{cnr} N_c}$$

$$\text{Lower Limits: } \frac{\text{Total relevant resources retrieved}}{\text{Total relevant resources retrieved} + U_{cnr} N_c}$$

$$\text{Upper Limits: } \frac{\text{Total relevant resources retrieved}}{\text{Total relevant resources retrieved} + L_{cnr} N_c}$$

Where p_{cnr} is the proportion of non-retrieved relevant resources found in the sample of the control set and N_c is the control set size. L_{cnr} and U_{cnr} are the lower and upper limits of the confidence interval for p_{cnr} which could be obtained from the formulas defined in the section 4.6.3.3.3.1

However, when the result set is also large and the sample must be used, the estimated recall could be defined as:

Estimated Recall = $\frac{p_d N_d}{p_d N_d + p_{cnr} N_c}$

Lower Limits: $\frac{L_d N_d}{L_d N_d + U_{cnr} N_c}$

Upper Limits: $\frac{U_d N_d}{U_d N_d + L_{cnr} N_c}$

Where p_d is the proportion of relevant resources found in the sample of the result set and N_d is the result set size. L_d and U_d are the lower and upper limits of the confidence interval for p_d which could be obtained from the formulas defined in the section 4.6.3.3.3.1.

Final estimated recall In obtaining a single number on the estimated recall, the average on the estimated recall obtained from both methods was used. The final confidence interval was defined in terms of the range where the confidence intervals of both methods overlapped. In particular, the highest lower limit of the two methods and the lowest upper limit of the two methods were used in defining the final confidence interval.

4.6.4 Results

4.6.4.1 Control Sets and Deduction Result Sets The results of the queries were obtained based on the data acquired from the collection via the Web service interface on March 21, 2004. The data was cached to provide the consistency of the data used across different queries. The data was preprocessed as necessary and was supplied to the deduction system. The deduction system produced the results for each query based on the supplied data and the query. The result sets of 63 queries (query number 1-63) were obtained. The result sets of 12 queries based on adhoc associations (query number 64-75) were subsequently obtained based on the result sets of the base queries. Because of the differences between the queries based on adhoc associations and the other deduction queries, the analysis of these result sets are provided separately in section 4.6.5.4. The results provided in this section and subsequent sections only include those of the 63 base queries.

The total number of resources in the control set and the result set for each query is listed in Table C.1 and C.2 in the Appendix C. The summary of the control sets and the result sets obtained for the queries are provided in Table 4.5. The numbers listed in the square brackets specify the query numbers.

Table 4.5. The Control Sets and the Deduction Result Sets Summary

Control Sets \ Deduction Result Sets	**0-25**	**26-100**	**101-1,000**	**>1,000**	**Total Sets**
0-500	10 [22,33,35,36, 37,38,39,51, 52,53]	3 [34,48,49]	1 [50]	-	**14**
501-1,000	5 [6,13,17,59, 60]	7 [16,55,56,57, 58,61,62]	7 [7,8,11,12, 14,54,63]	-	**19**
1,001-5,000	5 [10,19,26,45, 46]	8 [4,5,15,23, 25,32,41,44]	3 [9,21,47]	5 [20,27,40,42, 43]	**21**
>5,000	-	6 [1,2,3,18, 24,29]	1 [30]	2 [28,31]	**9**
Total Sets	**20**	**24**	**12**	**7**	**63**

4.6.4.2 Review Sets Fifty three result sets (84%) contained 500 or fewer resources. In these cases, all the resources in each set were selected for the relevance assessment. Ten result sets (16%) contained more than 500 resources. In these cases, 500 resources were randomly selected for each set for relevance assessment. Fourteen control sets (22%) contained 500 or fewer resources. In these cases, all the resources in each set were selected for the relevance assessment. Forty nine control sets (78%) contained more than 500 resources. In these cases, 500 resources were randomly selected for each set for the relevance assessment.

In preparing the resources for the relevance judgment, the selected resources of the control set and the result set for the query were combined into a single review set. This prevented the

same resource being reviewed twice for a query. The review set for each query was presented to the two judges for the assessment. Information about the resources, i.e. title, descriptions, table of contents, editorial reviews and excerpts, available from the collection was presented to the judges to assist in the judgment process. The judges independently reviewed and assessed the relevance of the resources in the review sets. The details on the relevant judgment tasks and tools are provided in the Appendix E.2.

4.6.4.3 Relevance Judgment Results The relevance judgments of the two judges were examined to find any disagreement in the judgment results. Out of the 31,940 resources in the review sets, 9,893 (31%) were assessed relevant by one or both judge. Of these, 4,849 (49%) were assessed relevant by both judges. 5,044 (51%) were assessed relevant by only one judge. These resources were quarantined and formed the review sets for the third judge. The third judge made a relevance judgment on these resources under the same judgment setting. Among these resources, the third judge assessed 3,438 (68%) resources as relevant. These resources, combined with those previously agreed as relevant, were considered the relevant resources for the queries. They were used as the basis for measuring and estimating the retrieval effectiveness of the result sets.

4.6.4.4 Precision and Recall in the Control and Result Sets The proportion of relevant resources found in each control set and result set is reported in Table C.1 in Appendix C. Based on the proportion, the total number of relevant resources, precision and recall of the control and result sets could be measured or estimated[1] as reported in Table C.2 and C.3.

The measured or estimated precision of each control set is plotted in Figure 4.5. Estimated precision is plotted along with the 95% confidence interval. Given that the recall of each control set by definition is equal to one, the recall plot for the control sets is omitted.

[1] See section 4.6.3.3.3 for the methods in estimating precision and recall of the large sets

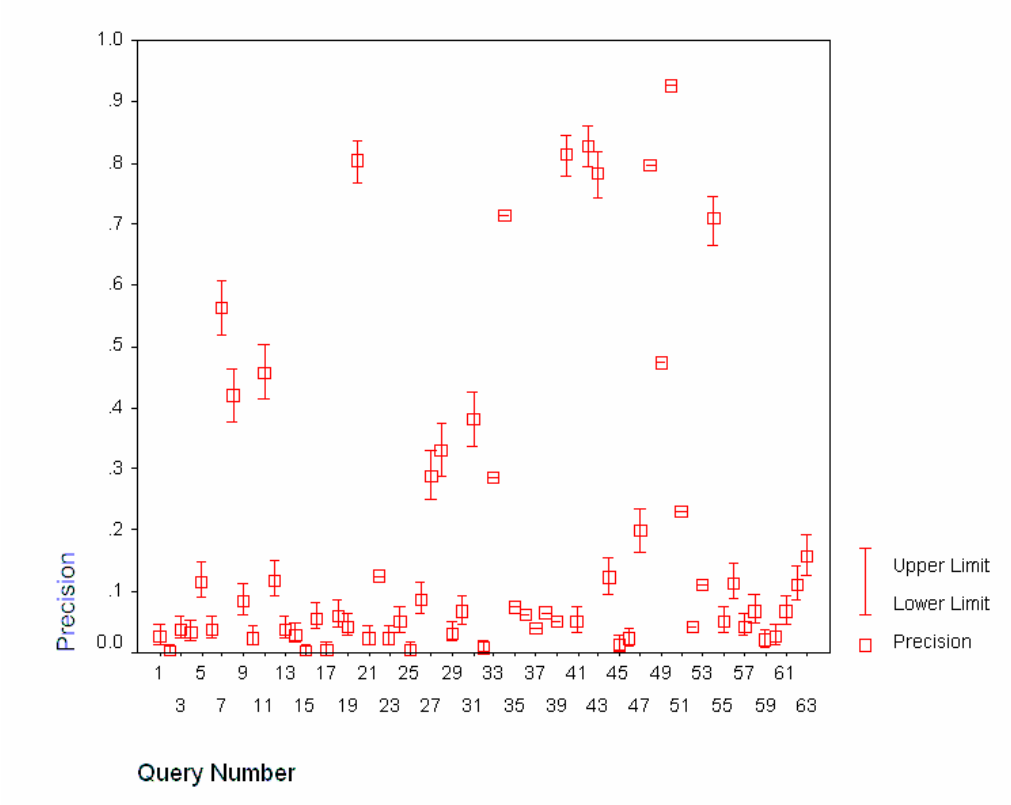


Figure 4.5. Measured and estimated precision of the control sets

The measured or estimated precision of each result set is plotted in Figure 4.6. Estimated precision is plotted along with the 95% confidence interval. The measured or estimated recall of each result set is plotted in Figure 4.7. Estimated recall is plotted along with the estimated confidence interval.

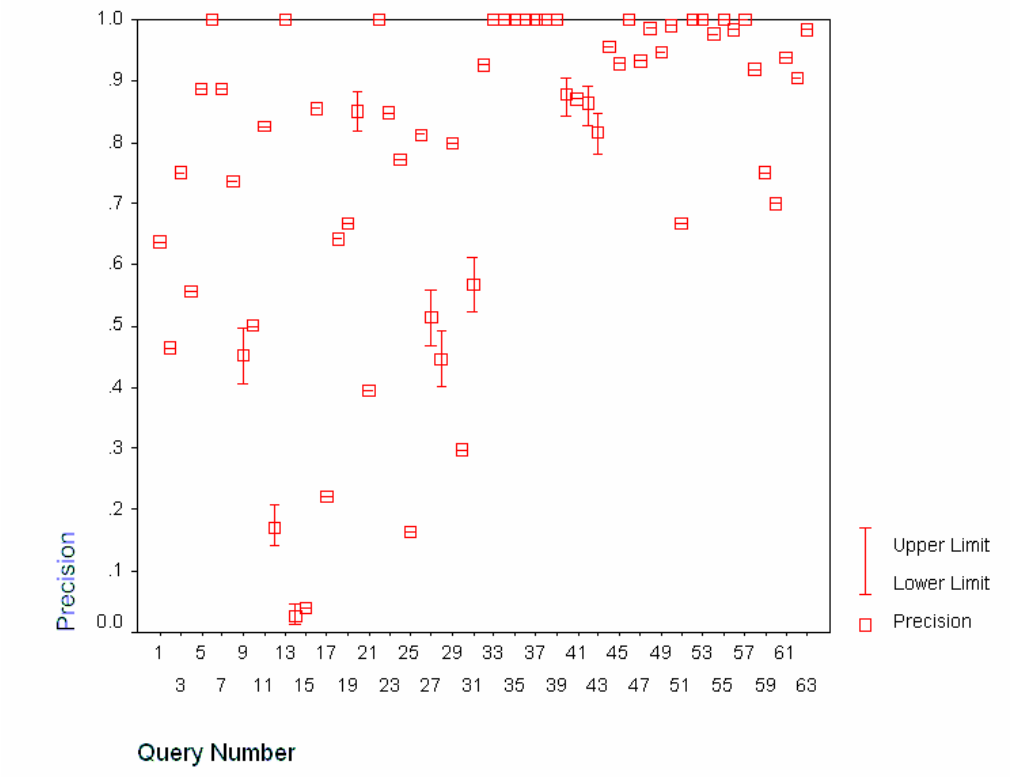


Figure 4.6. Measured and estimated precision of the result sets

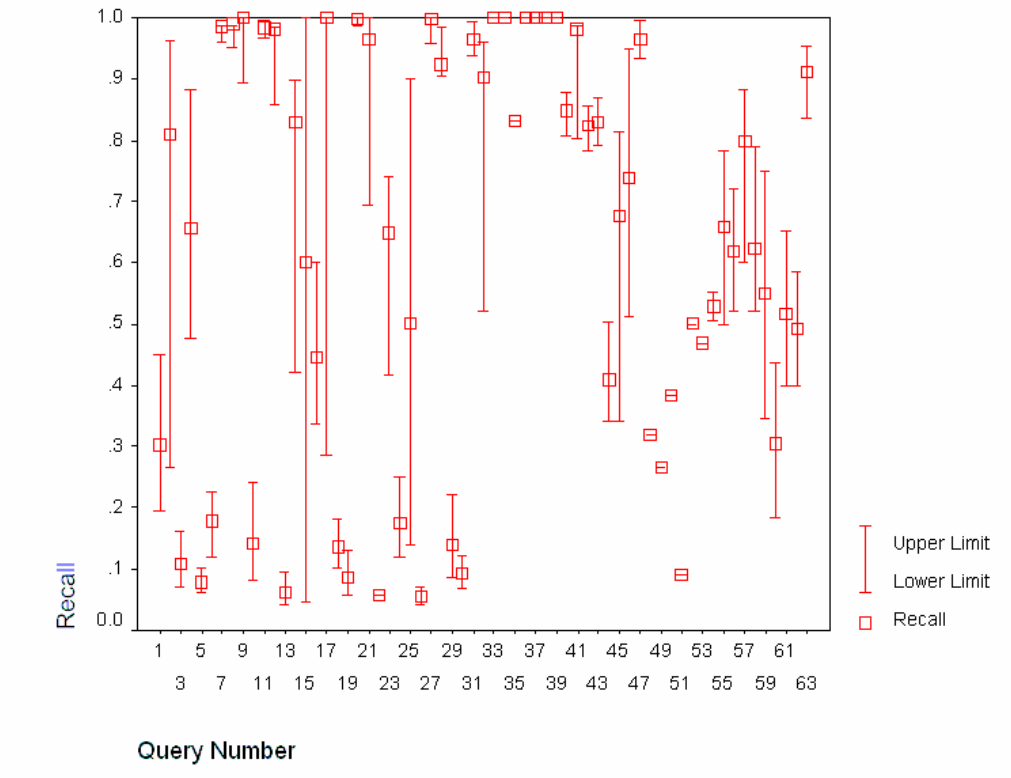


Figure 4.7. Measured and estimated recall of the result sets

The results show that the precision of 15 result sets (24%) were perfect. The precision of 20 result sets (32%) were less than one but were higher than 0.8. In addition, the precision of four result sets (6%) were estimated to be between 0.8 and one. In terms of recall, the results show that the recall of six result sets (10%) were perfect. The recall of one result set was between 0.8 and one. In addition, the recall of 21 result sets (33%) were estimated to be between 0.8 and one. The hypothesis testing results indicated that the null hypothesis of 15 result sets (24%) of having perfect precision could not be rejected, while the null hypothesis of 11 result sets (17%) of having perfect recall could not be rejected. The precision-recall of the results sets are summarized in Table 4.6

Table 4.6. Summary on precision-recall of the result sets

		Precision		
	0-0.5	**0.5-0.8**	**0.8-1**	**Total Sets**
Recall **0-0.5**	3 [10,25,30]	8 [1,3,18,19,24, 29,51,60]	13 [5,6,13,16,22, 26,44,48,49,50, 52,53,62]	**24**
0.5-0.8	1 [15]	2 [4,59]	8 [23,45,46,54,55, 56,58,61]	**11**
0.8-1.0	7 [2,9,12,14,17, 21,28]	3 [8,27,31]	18 [7,11,20,32,33, 34,35,36,37,38, 39,40,41,42,43, 47,57,63]	**28**
Total Sets	**11**	**13**	**39**	**63**

The precision and recall of the result sets are plotted against each other in the graph shown in Figure 4.8. In order to distinguish the result sets that were more effective from those less effective, the graph is divided into four major regions using the cut-off value of 0.8 in each axis. Each region approximately distinguishes the retrieval effectiveness of the result sets as follows. The region in the upper right corner of the graph indicates the most effective retrieval, i.e. high precision/high recall (precision>0.8, recall>0.8), of the result sets. The retrieval effectiveness of 18 result sets (29%) fell in this region. The region in the lower right corner of the graph indicates

the high precision and low recall of the result sets (precision>0.8, recall<0.8). Those of 21 result sets (33%) fell in this region. The region in the upper left corner of the graph indicates the low precision and high recall of the result sets (precision<0.8, recall>0.8). Those of 10 result sets (16%) fell in this region. The region in the lower left corner of the graph indicates the least effective retrieval, i.e. low precision/low recall (precision<0.8, recall<0.8), of the result sets. Those of 14 result sets (22%) fell in this region. It should be noted that the six result sets whose precision and recall are perfect are displayed as a single point in the graph.

The precision-recall plot for the control sets resulted in the points scattered along the top horizontal line of the graph (recall = 1). Fifty nine control sets (94%) fell under the low precision/ high recall area. Four control sets (6%) fell under the high precision/high recall area. The plot is omitted for brevity.

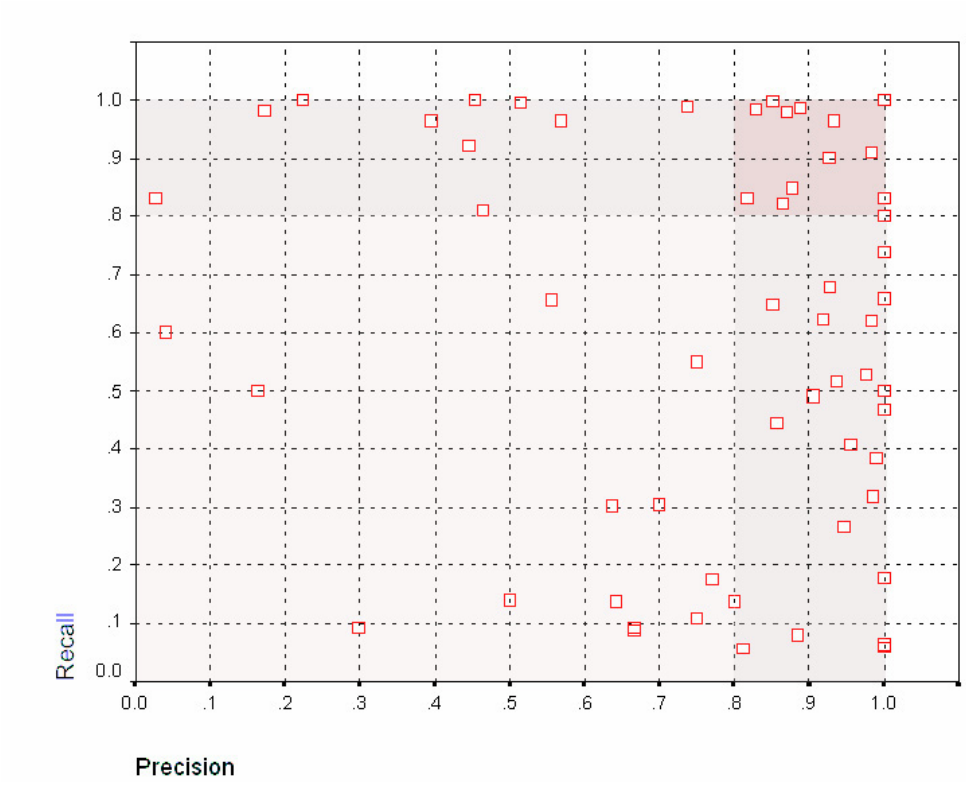


Figure 4.8. Precision-recall plot for the result sets

4.6.5 Analysis of Results

4.6.5.1 Deduction Impact Analysis One way to provide a simple indication of the impact of deduction is to look at how it changes overall precision and recall of the control sets. Given that the recall within the control set is always equal to one, the recall of the result set is always less than or equal to that of the control set. The precision of the control set will be less than one. The precision of the result set can be greater or less than that of the control set. The changes in precision and recall of the control sets when deduction applied are plotted in Figure 4.9. If the system results were perfect, all lines would be horizontal vectors moving right.

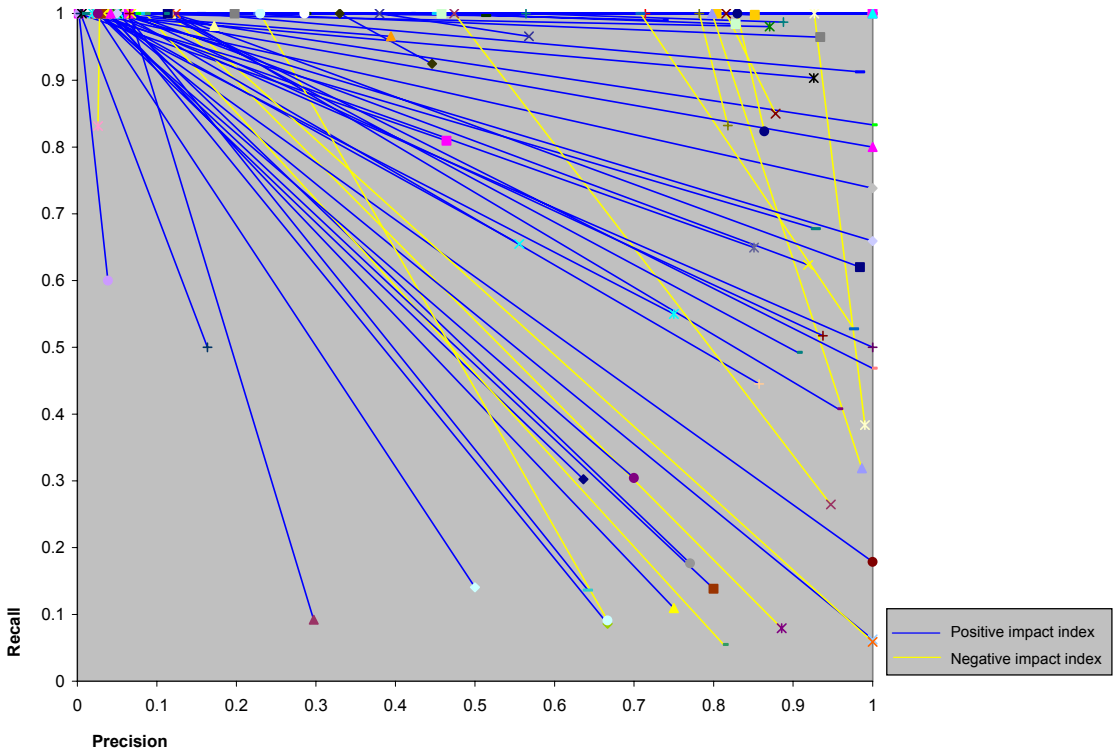


Figure 4.9. The changes in precision and recall in the control sets

To measure overall impact of deduction on a control set, change in the harmonic mean of precision/ recall (*F-measure*[1]) is measured as the *impact index*. The deduction impact index for each control set is reported in Table C.2. In summary, the positive index implies the increase in precision outweighs the decrease in recall, while the negative index implies the decrease in recall

[1] F-measure is a common combined measure in evaluating retrieval performance that is defined based on E-measure [131]. F-measure= 2*Precision*Recall/(Precision+Recall)

outweighs the increase in precision or both decreased. The largest impact index value was 0.925 for query 37. The smallest was -0.409 for query 50. Fifty one queries (81%) were positive and 12 queries (19%) were negative. The average impact index across all queries was 0.328.

In order to identify the impact of deduction by queries, the subject areas of the queries are grouped into nine major categories: computers/technologies (9 queries), US president biography (10 queries), other biography/history (3 queries), travel (7 queries), art history (7 queries), science fictions (9 queries), other fictions (6 queries), seafood cooking (6 queries) and other cooking (6 queries). The changes in precision and recall in the control sets are plotted by subject area as shown in Figure 4.10.

The average impact index for the queries in each subject area is summarized as follows.

- computers/technologies (0.507)
- US president biography (0.516)
- other biography/history (0.198)
- travel (0.137)
- art history (0.125)
- science fictions (0.657)
- other fictions (0.202)
- seafood cooking (-0.039)
- other cooking (0.268)

It should be noted that the deduction impact analysis shows the retrieval effectiveness of the result sets in the context of the control sets. It also allows the defined queries to be assessed by subject areas. The investigation on the factors contributing to degraded retrieval performance will be identified in the following analyses.

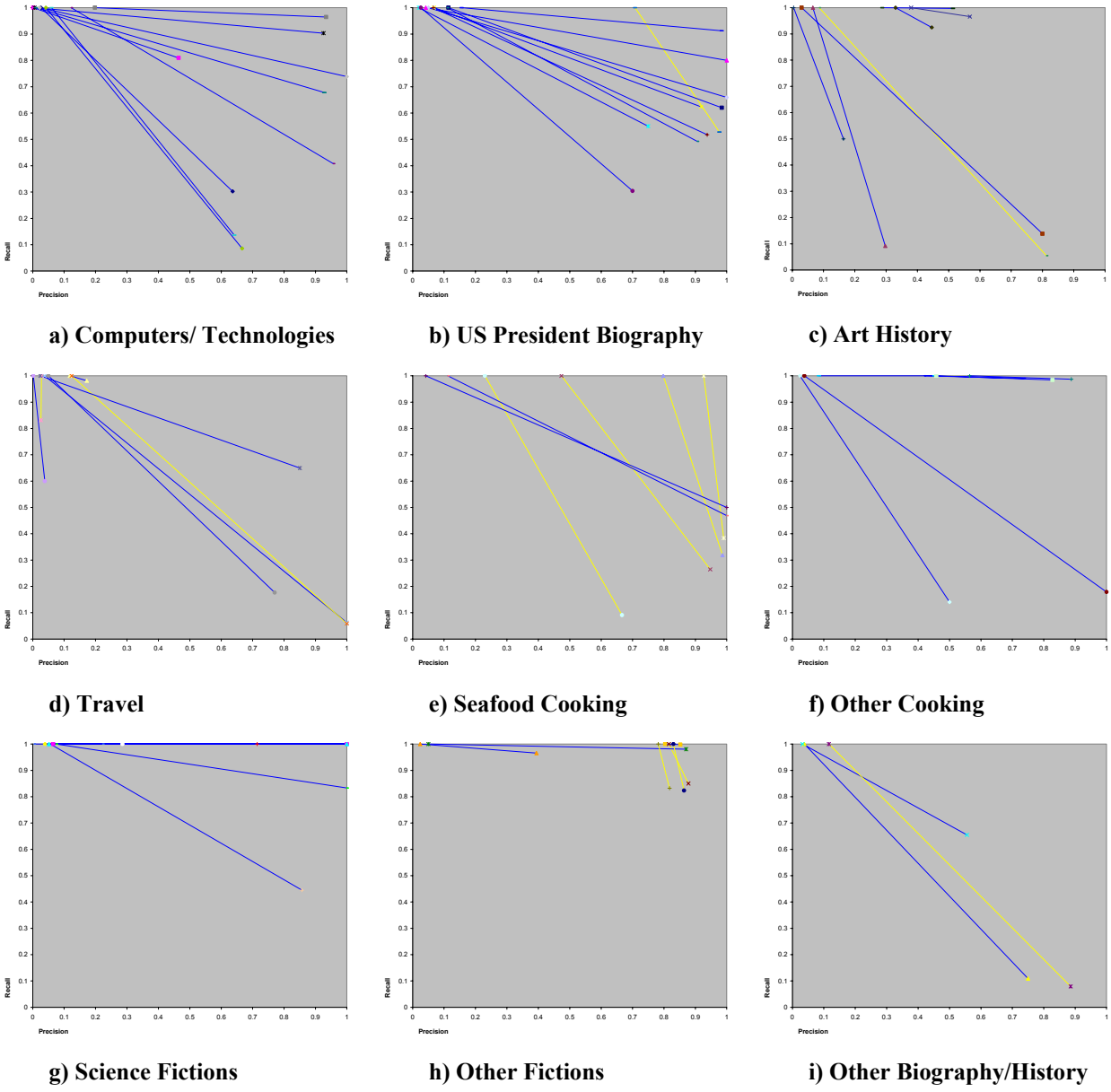


Figure 4.10. The changes in precision and recall in the control sets by subject area

4.6.5.2 Causes of Degraded Precision and Recall in the Result Sets The result set assessment found the retrieval performance of the deduction system less than perfect. This section describes the factors contributing to degraded retrieval performance. In particular, it discusses the major causes of non-relevant resources being retrieved by the system i.e. "false positives", and relevant resources not being retrieved by the system i.e. "misses". The explanation is based on observations made on the relevance judgment results and some informal interviews with the judges.

Contributing factors to the degraded retrieval performance were found to be limitations of the test collection and the study settings. The factors could be classified into six major categories. Three factors are related to the collection and include misclassification of resources, inconsistency in subject classification and omission of information. Two factors are related to the study and include inaccuracy and inadequacy of query expressions, incomplete semantic coverage in the added semantics. The last factor is a mismatch in the level of specificity between the collection and the study. These can be described in details as follows.

4.6.5.2.1 Misclassification of resources The misclassification of resources degraded precision of the result sets. Misclassified resources are the resources assigned to the categories where they do not belong. For example, it was found that some travel books on the Holland Counties in Ohio, Michigan and Pennsylvania were classified into the subject category of the travel books on the country Netherlands. Further, some books on the historic New Holland[1] voyages were also found under the category of the travel books on the country Netherlands. Misclassified resources were found periodically in different categories across subject areas. For example, it was found that some military books were misclassified as war fiction. These were likely caused by resources being classified based on text in their titles. The inaccuracies resulted in degraded precision for the queries involving the categories with misclassified resources (e.g. queries 12-15, 40-43, etc.).

In some cases, it was found that misclassified resources were those closely related to the members of the categories. For example, the biography books on the first ladies were sometimes assigned into the same categories as those of the US presidents. The books on the impacts of the Iraq-Iran war on other countries were assigned into the categories of history of Iraq/ Iran. Although these resources might have some indirect relevance to the categories, they could not be considered members of the categories. It is possible that these resources were placed in the given category because a more appropriate category did not exist. The inaccuracies caused by inappropriate category members degraded precision for the queries involving the categories with such resources (e.g. queries 4, 5, etc.).

[1] New Holland in the historic voyage was the name given to Australia by the Dutch

4.6.5.2.2 Inconsistency in subject classification The assignment of subcategories has an impact on the precision of the result sets. In particular, some inconsistent assignments of subcategories led to degraded precision. For example, the category "Computers& Internet/ Databases/ Data Storage & Management" category includes the subcategories: Data Mining, Distributed Computing and Encryption along with others. Although these subject categories relate to the parent categories to some degree, semantically they are not subsumed by their parent categories. More specifically, the members of the subcategories can not be considered the members of the parent categories. This resulted in degraded precision for the queries involving the parent categories with inappropriate subcategory assignment (e.g. queries 1, 2, 18, etc.).

In some occasions, degraded precision could also be caused by the subcategories that are partially subsumed by their parent directories. Specifically, some members of these categories could be considered the members of the parent category while some could not. For example, the category "History/ Americas/ United States/ 19th Century" contains a subcategory "Turn of the Century". It was found that the subcategory contains the books on the US history at the end of the 19th century as well as those on the US history at the beginning of the 20th century. For instance, books on US history during the presidency of the former US president Theodore Roosevelt (1901-1909) were assigned to the category but could not be considered books on US history in the 19th century. Although it is sometimes difficult to distinguish partially relevant subcategories from those with misclassified resources, the focus here is on the integrity of subsumption in the subject category hierarchy. This resulted in degraded precision for the queries involving the parent categories with such subcategories assignment (e.g. query 5).

4.6.5.2.3 Omission of Information Omission of information degraded recall of the result sets. Omission is different from misclassification in that, with omission, resources could be classified into the proper subject categories but may not be classified into every subject category they apply to. Information omissions were present in several forms. The most common form is the lack of classifying dimensions of the resources. For examples, some books on the history of Mesopotamia were not included in the subject category of history on Iraq. Books on the history of Flemish arts were not included in the subject category of Dutch history. Books on US history during the presidency of Abraham Lincoln were not included in the subject of US history in the

19th century. This resulted in degraded recall for the queries involving categories with omitted resources (e.g. queries 4, 5, 26, etc.).

The information could also be omitted by being given too generally. The lack of the specificity of the given information contributed to partial omissions. For examples, some biography books on the US presidents were classified using the subject terms Presidents and Head of States rather than using the particular names of the persons. Cooking books on a specific kind of fish were classified using the subject term Seafood instead of the particular kind of fish. This form degraded the recall when the queries related to the specific information omitted (e.g. queries 48-63).

Although information omissions usually do not degrade the precision of the result sets, there is one exception. When the queries involved negation and the closed-world assumption is used, omissions could have the major impact on the precision on these result sets. With incomplete information, incorrect inferences will be made related to the non-members of the categories. Such omissions result in degraded precision for the queries involving negation (e.g. queries 7, 8, 9, 11, 17, 27 etc.).

4.6.5.2.4 Inaccuracy and inadequacy of query expressions As indicated in the introduction to this section, two aspects of the study design were found to have some impacts on the results. The first related to the translation of the English language query into specific query expression. The accuracy and adequacy of the query expressions formulated had a severe impact on both the precision and recall of a few result sets in this experiment. Insufficiency of the query expressions resulted in degraded recall. For example, the query on *books on security in electronic commerce* (query 19) did not refer to the *Network Security* and the *Online Privacy* subject categories in the query expression in addition to the *Encryption* and the *Cryptography* subject categories. Thus, the relevant resources existing in the two omitted subject categories were not retrieved by the deduction system. This resulted in degraded recall of the result set for the query. In addition, misrepresented query expressions could result in degraded precision. For example, the query on *books on the history of piano* (query 25) was formulated based on the subject category "*piano*", which implies not only the instruments but also the performers, music pieces and notes. Thus, the

formulated query expression should in fact represent *books on the history of piano music, instruments and performers* rather than the history of the piano as instrument. The inaccuracy in formulating query expression led to the retrieval of the resources non-relevant to the query.

4.6.5.2.5 Incomplete semantics coverage in the added semantics The recall of some result sets was deteriorated due to the lack in the coverage of the added semantics. For examples, the media formats *Mass Market Paperback* and *Library Binding*, which are two specific kinds of media format, were unforeseen and were not included in the added semantics on media formats. Thus, the resources in these formats were not retrieved as those in printed formats as required by some queries (i.e. query 40, 42 and 43). The branches of the Prentice-Hall publishing company, which include the specific publishing divisions such as Prentice-Hall Professional Technical Reference (PTR) and Prentice Hall College Division, were unforeseen and were not modeled as specific kinds of the Prentice Hall publishing company. Thus, the system failed to retrieve the relevant resources related to them (i.e. query 44). It was also found that relevant resources on the cooking of parts and variations of fish, i.e. Caviar, Sashimi (Japanese raw seafood) were not retrieved partially due to the lack of coverage by the added Seafood cookery semantics. These have contributed to degraded recall in some result sets related to them (i.e. query 48-53).

4.6.5.2.6 Mismatches in level of specificity The resources having more general or more specific content than that required by the queries also contributed to degraded retrieval performance. In particular, some resources were judged as non-relevant due to generalized or specialized content, even though these resources may be partially relevant to the queries. For example, the books on painters were not considered relevant to queries on art history because they were considered more specific than those required by the queries. Similarly, maps of the cities were not considered relevant to the queries on maps of the countries, which they are parts of, because they were considered more specific. Books on French cooking including some dessert recipes were not considered relevant to the queries on books about French desserts due to their generality. This resulted in degraded precision for some queries such as those on the travel, art history and cooking subject areas (e.g. queries 6, 12-15, 23-24 and 26-31, etc.).

4.6.5.2.7 Others Other factors were defined as controlled by the experimental design and were not considered contributing factors. However, these controls could be imperfect and have some impacts on the assessment results. In particular, these include relevance judgment errors, sampling errors and inaccuracies in data acquisition. Relevance judgment errors were mistakes made by human judges. Such impacts should be reduced in the future experiments by using more judges. This will help to ensure that the errors made by few judges will be compensated by others. Sampling errors occur when the measurement on the sample does not provide a good estimation of the actual set. Such an impact should be reduced in the future experiments by using a larger sample. The reliability of data acquisition, i.e. using the Webservice interface, could also impact the accuracy of the results. In particular, if there were errors in the supplied data by the data source, the results of the study would be less accurate. The reliability in providing the data, i.e. that it provides the consistent capture of the collection, must be ensured by the data source in order to maximize the accuracy of the results.

4.6.5.3 Sensitivity Analysis This section discusses the sensitivity analysis. A sensitivity analysis was conducted to observe some patterns which might impact the retrieval effectiveness of the deduction system. The analysis was conducted on three major aspects of the retrieval process: resources, subject categories and query expressions. The variables under the investigation are the number of resources, the number of relevant resources, the category structure and the query expressiveness. The relationships between these variables and retrieval effectiveness are examined. In particular, a chi-square test of independence was used in the tests at the significance level (p-value) of 0.05. The tau-b (τ_b) measure is used as a measure for the association strength. The dependent variables of the tests are precision and recall. The precision of the result sets are classified into two groups: those above 0.8 and those equal to or below 0.8. Similarly, the recall of the result sets are classified into two groups: those above 0.8 and those equal to or below 0.8. The complete results of the chi-square tests are reported in Appendix D. A summary of the test results and a discussion of some implications of the results are provided as follows.

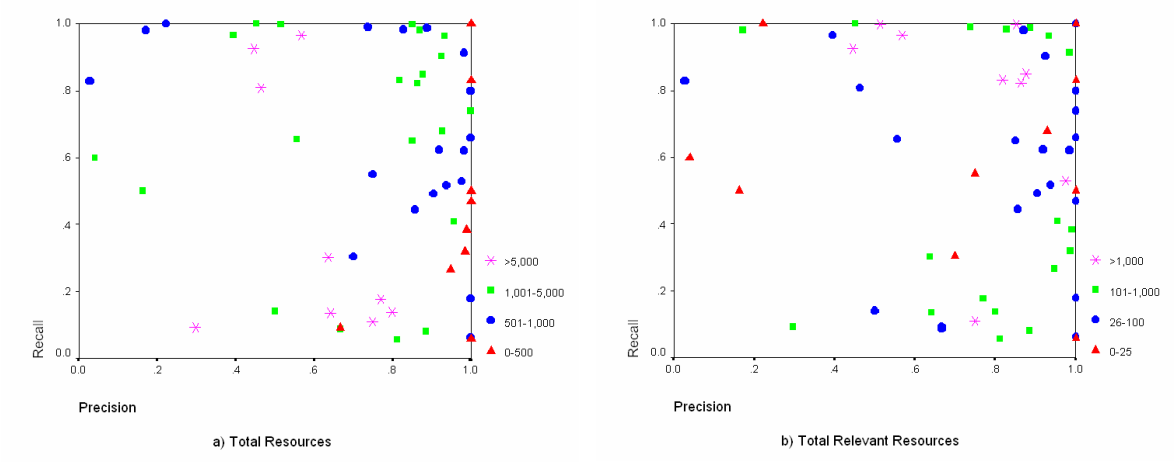


Figure 4.11. Precision-recall plots grouped by resource volume

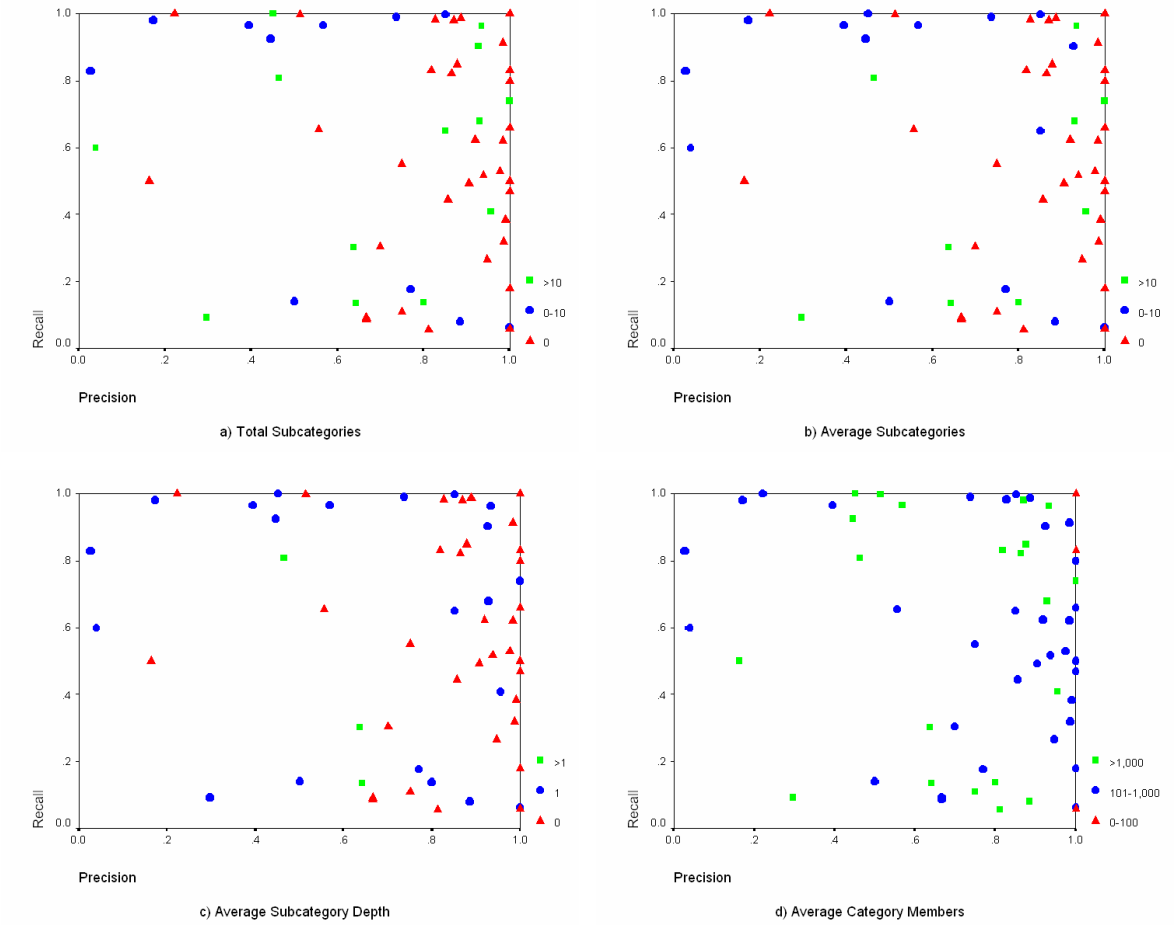


Figure 4.12. Precision-recall plots grouped by subject category properties

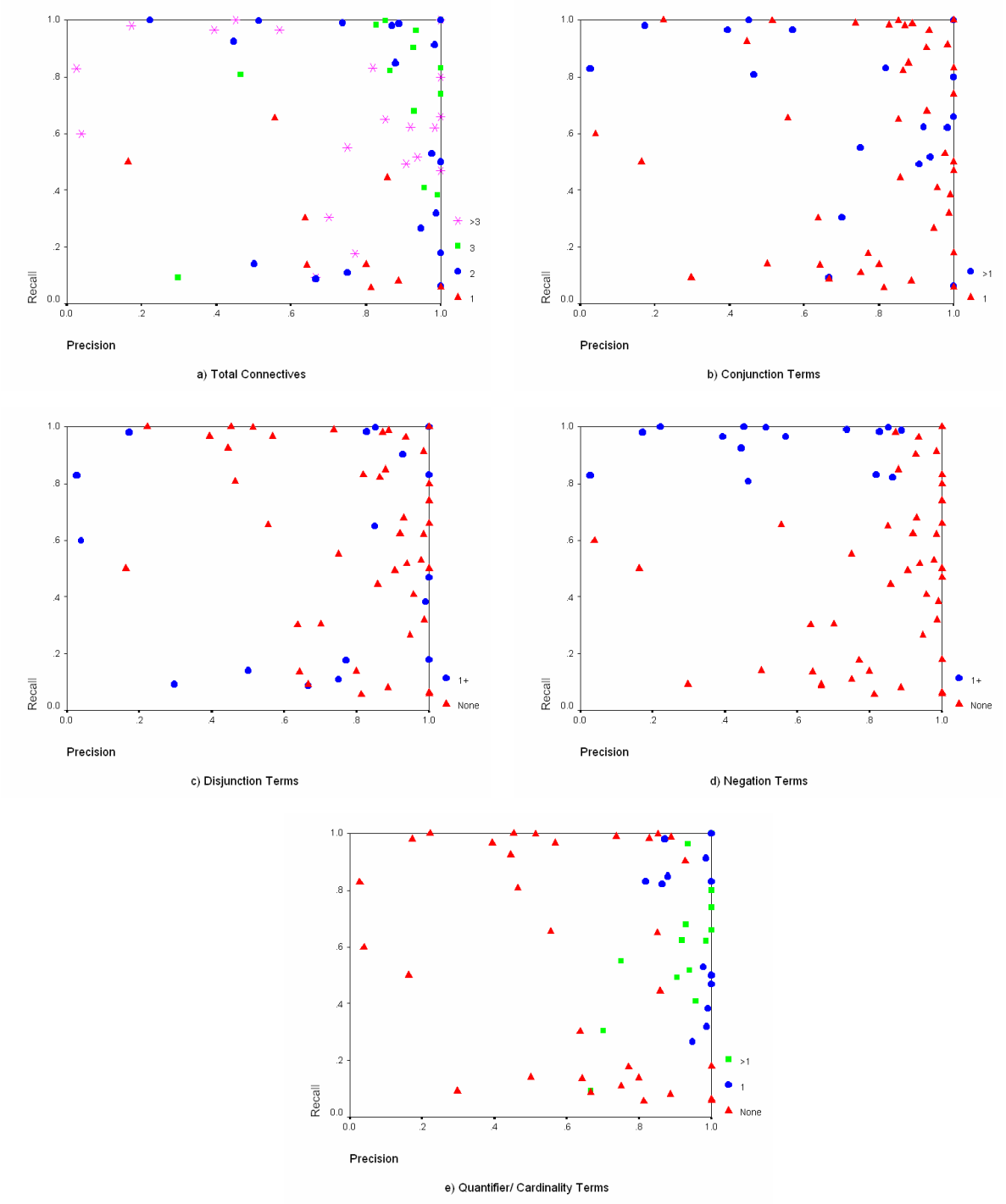


Figure 4.13. Precision-recall plots grouped by query expressiveness

The first aspect of the sensitivity analysis relates to the number of processed resources and the number of relevant resources. The precision-recall plots of the result sets grouped by these variables are shown in Figure 4.11a-b. The chi-square tests were conducted on these variables against precision and recall of the result sets. The results show an evidence of association between the number of resources processed and the precision of the result sets (p-value = 0.000) but not recall (p-value = 0.861). In particular, when a smaller number of resources was involved, the result sets exhibits a tendency for better precision (τ_b = -0.458). A simple explanation would be that, when larger number of resources is involved, inaccuracies in the resource information could be more likely introduced. This can result in degraded precision. The number of relevant resources does not exhibit relationship with precision (p-value = 0.834) or recall (p-value = 0.118). Thus, the retrieval performance of the deduction system was not found to be dependent on whether there are more or less relevant resources to be retrieved.

The second aspect of the sensitivity analysis involves subject category structure. In particular, the total and average number of subcategories and the average maximum subcategory depth of the categories involved in a query were examined. In addition, the average category size of the categories involved in a query was also examined. The precision-recall plots of the result sets grouped by these variables are shown in Figure 4.12a-d. The chi-square tests were conducted on these variables against precision and recall of the result sets. The results show some evidences that the queries involving the subject categories with simpler subcategory structure have better precision than those with more complex structure (p-value = 0.004, τ_b = -0.340 on total subcategories, p-value = 0.006, τ_b = -0.354 on average subcategories, p-value = 0.003, τ_b = -0.410 on average maximum subcategory depth). A simple explanation would be that when a subject category contains no subcategory, there was no impact from inconsistency in subcategory assignment. When subject category has more complex subcategory structure, inconsistency in subcategory assignment can have more impact on the accuracies of the categories above them.

The results also show some association between the average category size and precision (p-value = 0.034) and recall (p-value= 0.017). In particular, when the average category size is smaller, the result sets show a tendency for better precision (τ_b = -0.282) and recall (τ_b = -0.102).

A simple explanation would be that smaller categories are more narrowly defined and thus less susceptible to incorrect classification compared to larger categories.

The third aspect of the sensitivity analysis involves query expressiveness. The effects of query expressiveness are examined in terms of the number of connectives used in the expression, the occurrences of conjunction terms, disjunction terms, negation terms and quantifier/cardinality terms in the expression. The precision-recall plots of the result sets grouped by these variables are shown in Figure 4.13a-e. The chi-square tests were conducted on these variables against precision and recall of the result sets. The results show an association between the number of connectives used in the expression and recall (p-value = 0.019) but not precision (p-value = 0.148). In particular, when queries were expressed more verbosely, there was a tendency for better recall (τ_b = 0.152). A simple explanation would be that when queries were described expressively and sufficiently, there was a better chance that relevant resources will be included for selection.

In terms of the utilized expressive power, the results show no association between the use of conjunction and precision (p-value = 0.319) or recall (p-value = 0.390). The results also show no association between the use of disjunction and precision (p-value = 0.667) or recall (p-value = 0.759). However, the results show some association between the use of negation and precision (p-value = 0.009) and recall (p-value = 0.000). In particular, when negation is used, the results show a tendency for a decrease in precision (τ_b = -0.329) and an increase in recall (τ_b = 0.625). A simple explanation would be that the queries involving negation are less precise, and thus are more susceptible to more resources being retrieved than needed. Thus, the recall is usually high while precision can be varied. Further, the retrieval for the queries involving negation using the closed-world assumption is highly susceptible to the omission of category members. Thus, precision could be easily degraded by it. The results also show an association between the use of quantifier/ cardinality and precision (p-value = 0.000, τ_b = 0.468) but not recall (p-value = 0.133). A simple explanation would be that the information sought by the queries involving quantifier and cardinality is fairly straightforward and is less susceptible to inaccuracies.

4.6.5.4 Judge Agreement in the Relevance Judgment In this section, the degree of judge agreement in relevance judgment is examined. When the judges made the relevance judgment on a resource, the judges *agreed* if the resource was assessed as relevant by both judges. The judges *disagreed* when the resource was considered relevant by only a single judge but not by the other. In this analysis, the proportion of the *agreed* resources to the sum of those *agreed* and *disagreed* is called the Relevance Agreement ratio (*Ra*).

For each query, *Ra* could range from 0 to 1.When both judges select the same set of relevant resources, *Ra* will be maximized at 1. When both judges select the entirely different sets of relevant resources, *Ra* will be minimized at 0. *Ra* measured for the review set of each query is reported in Table E.1 in Appendix E. The measured *Ra* is summarized in Figure 4.14.

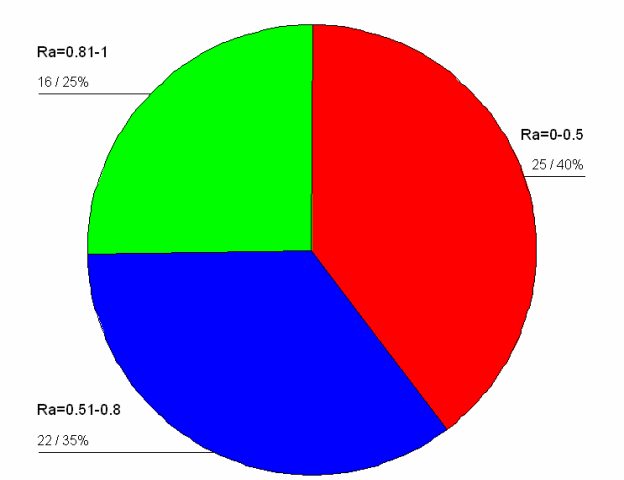


Figure 4.14. Summary of the measured relevance agreement ratio for the review sets

In summary, the review sets of 16 queries (25%) have *Ra* above 0.8; those of 22 queries (35%) have *Ra* between 0.5-0.8 and those of 25 queries (40%) have *Ra* below 0.5. In order to examine the relationship between the degree of judge agreement and the retrieval effectiveness, the sensitivity analysis is conducted on *Ra* against precision and recall of the result sets. The precision-recall plot of the result sets grouped by the judge agreement degree is shown in Figure 4.15.

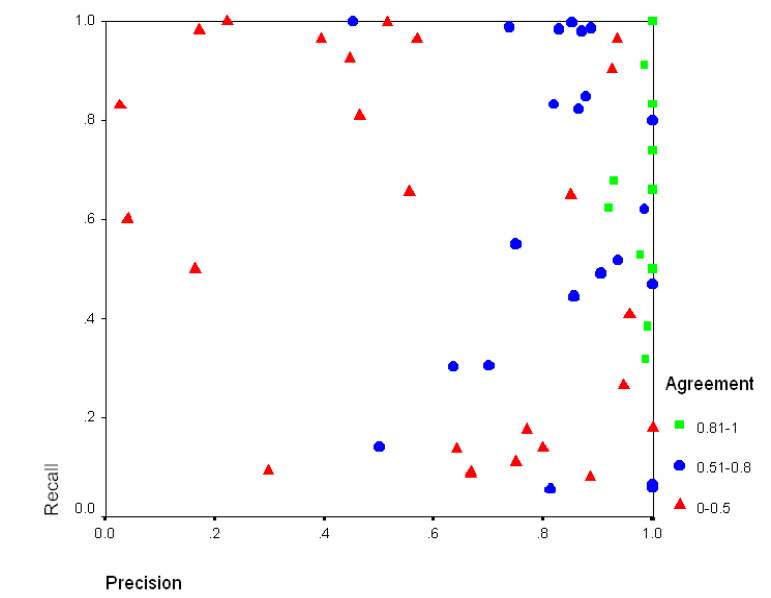


Figure 4.15. Precision-recall plot grouped by judge agreement degree

The results from the chi-square test show an association between the degree of judge agreement and precision (p-value = 0.000) but not recall (p-value = 0.815). In particular, when the judges could agree more on the relevancy of the resources, the result sets exhibit a tendency for better precision ($\tau_b = 0.571$).

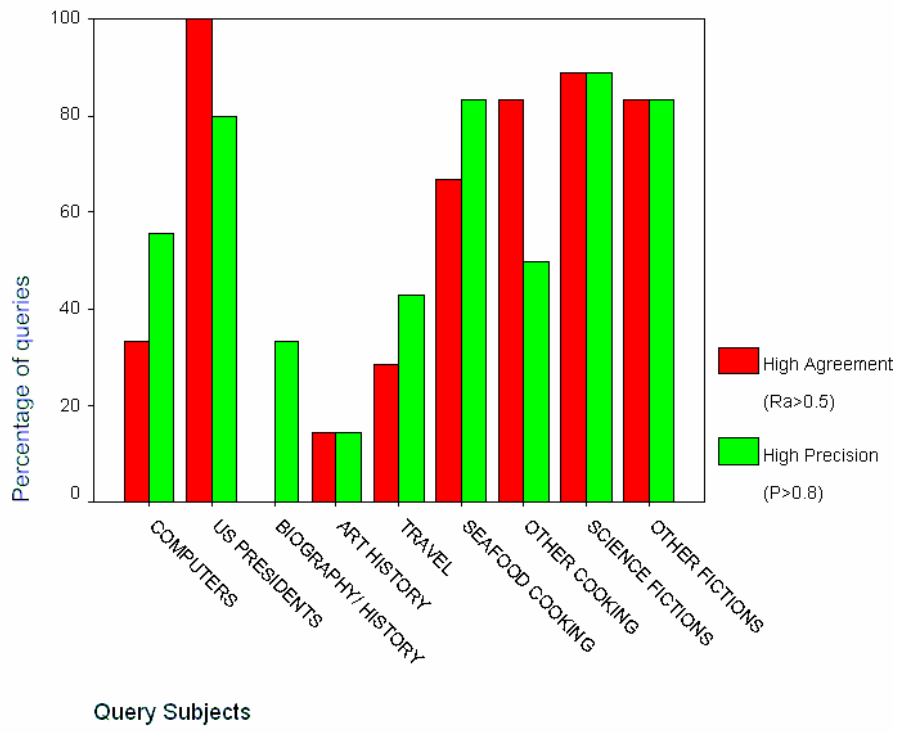


Figure 4.16. Proportion of queries with high relevance agreement/ high precision by subject areas

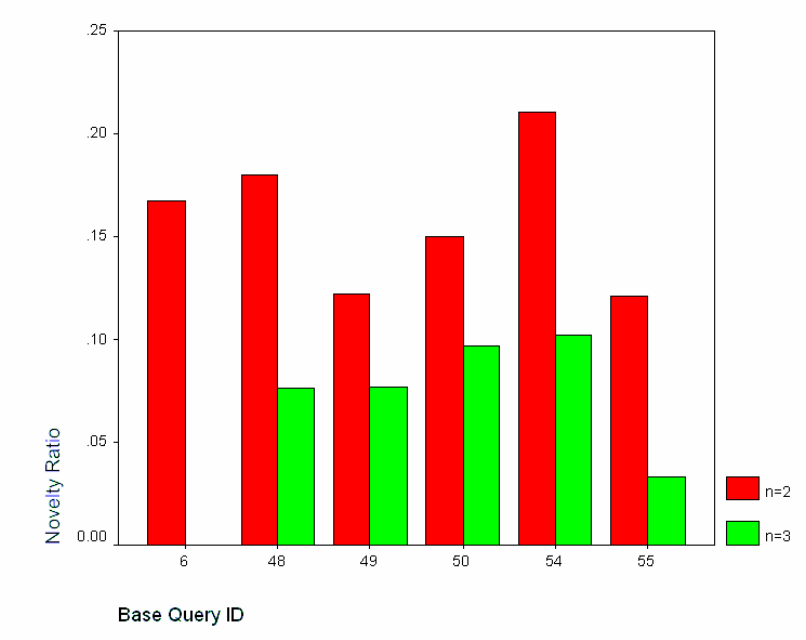
A further analysis based on the query subject areas show that the judges could agree more on some particular subjects over another (Figure 4.16). For example, the judges could agree more about the resources relevant to the queries on US president biography than those on art history. Precision also shows the similar trend for these subjects, i.e. the queries on US president biography has overall better precision than those on art history. A simple explanation would be that individual perception on the semantics of art history, which could involve paintings, sculptures, architectures, photography, artists, exhibitions, decorations, etc., could be more diverse in comparison to those on the US presidents. When the query subject is simpler, retrieval accuracy will be less likely impacted by individual differences in semantics perception.

4.6.5.5 Assessment on the Retrieval Effectiveness of the Queries using Adhoc Associations

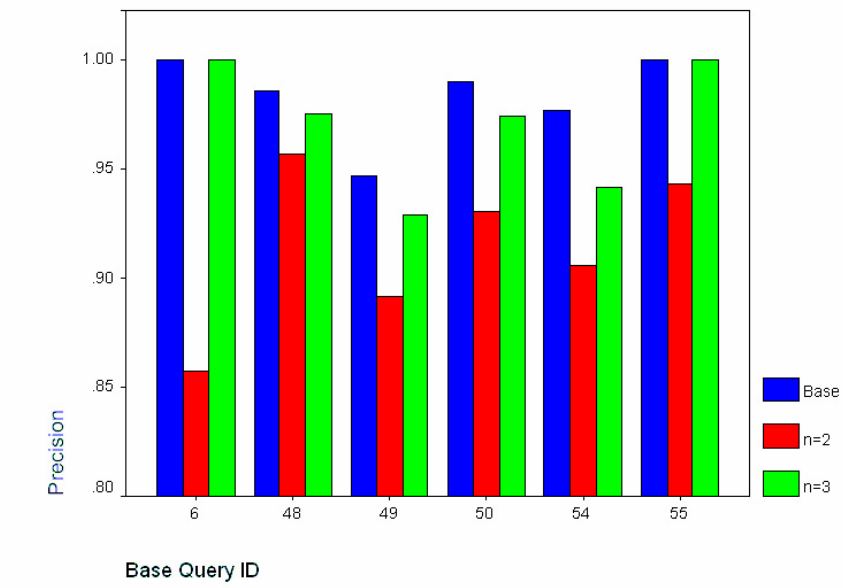
An assessment was made of the use of adhoc association to improve retrieval performance. In particular, the impacts on the result sets in terms of novelty ratio and precision were measured. An independent variable in the assessment was the association degree (*n*) used in query. 12 queries were used in the assessment, i.e. query number 64-75 listed in Table B.1 in Appendix B.1. The queries were defined based on six base queries. Specifically, two adhoc queries were created per base query. Each query utilizes the adhoc association with a different association degree (n=2 or 3). It should be noted that the result set of base query is always a subset of the result set of adhoc query defined based on it.

The queries using adhoc association were run against the deduction system to produce new result sets. All the obtained result sets were small, i.e. each contains less than 500 resources. Thus, all the resources in each result set were reviewed and assessed for their relevancy. The relevance judgments were made on the resources similar to those conducted for other queries. The judge agreement in relevance judgement was high for all the result sets, i.e. *Ra* was above 0.8 in each result set. Based on the number of relevant resources obtained for each result set, the novelty ratio and precision of the result sets were measured. In particular, novelty ratio was the proportion of the new relevant resources retrieved, i.e. those not included in the result set of the base query, to the total relevant resources retrieved. Precision was the proportion of the relevant resources retrieved to the total resources retrieved. The assessment of the adhoc result sets is provided, compared with those of the base queries, in Table C.4 in Appendix C. The novelty

ratio of the result sets is plotted in Figure 4.17a. The precision of the result sets is shown in Figure 4.17b by comparing it with that of the base queries.



a) Novelty ratio of the result sets



b) Precision of the result sets

Figure 4.17. Retrieval effectiveness for the queries using adhoc associations

The results show the impact of the selected association degree (n) on the novelty ratio and precision of the result sets. In particular, the result sets show better novelty ratio when the

smaller *n* is used, while the result sets show better precision when the larger number of *n* is used. The tradeoff in the novelty ratio and precision was present in all the result sets and could be explained as follows. When lower *n* is used, more resources could be retrieved by means of the adhoc associations. These include relevant as well as non-relevant, thus this usually results in a better novelty ratio and worse precision. When higher *n* is used, fewer resources were retrieved by means of the adhoc associations. Thus, fewer relevant resources as well as fewer non-relevant resources were retrieved. This results in a smaller increase in novelty ratio and a smaller decrease in precision. However, if the selected *n* is too large, no new resource will be retrieved. Inversely, if the selected *n* is too small, the new resources retrieved will be close to random, which could dramatically reduce the precision of the result set.

It should be noted the assessment was intended as a preliminary study. Its major goal was to identify the potential of deduction over adhoc associations and to provide preliminary assessment of the impact on the retrieval performance. The assessment was simplified in various aspects. For instance, the base queries chosen for the assessment were relatively broad queries, which enabled more resources to be associated with the base result sets. The number of associations employed for each resource was limited to five in order to limit the computational complexity. This limited the value of *n* that could be applied to the query. Further investigation and assessment could be conducted using variations of the technique in more complex settings.

4.7 CONCLUSION

Although individual standards and technologies for the Semantic Web are emerging, an intregrated framework and system that demonstrates the potential of the Semantic Web in the organization and discovery of information resources is still lacking. The case study was a research effort investigating the use of the Semantic Web technologies for the finding of resources in a real-world setting. The results of this research indicated potentials of these technologies in supplementing the finding of resources. The results also suggested some factors that can impact on the performance of the deduction system. Further investigation will be required to address more complex issues and improvement.

The case study demonstrated the positive impacts a deduction system can have on the retrieval of resources. With the deduction system, the overall retrieval performance in the collection was improved over the defined queries (overall impact index = +0.328). The case study demonstrated the uses of the deduction system for the retrieval of resources based on semantics in decentralized fashion. It demonstrated the deduction techniques that can be applied over metadata and ontologies and the queries that can be composed based on the available semantics. An adjunct preliminary study suggests the potential positive impact of deduction applied over adhoc associations. Results indicated that deduction can have a positive impact not only on classified resources but on resources gathered through associations.

The study highlighted factors that could degrade the retrieval performance of deduction systems. Errors and omissions in semantics representation were found to impact on the performance of deduction system. Inaccuracies in metadata and ontologies were found to be associated with complexity in the processed semantics, i.e. resource volume and subcategory assignments. Some particular uses of expressive power, i.e., negation, were found to be a likely cause of volatile retrieval accuracy. The queries involved with complex subjects were found more susceptible to individual perspective and can impact the retrieval accuracy.

The study has succeeded in demonstrating the overall positive impact of the uses of deduction techniques applied over metadata and ontologies. Further research will be required for additional improvement in their effectiveness.

4.8 RECOMMENDATIONS FOR FUTURE RESEARCH

The retrieval of some result sets was degraded by misclassified resources. One of the possible solutions is to combine an automated mechanism of identifying misclassified resources. Future research may investigate text analysis techniques that can be used to identify those resources that are potentially misclassified and prevent their inclusion. Ranking of the results will allow the retrieval of resources based on degree of relevance. Future research may investigate some

metrics used in examining degree of relevance of resources to a query. Individual differences in semantic interpretation should be further investigated. It became apparent from the results that not all semantics were viewed as equally clear by the judges. Some mechanism will be needed to identify and compensate for "fuzzy" semantics. Using adhoc associations to improve retrieval performance should be explored in a larger context. Future research may investigate a more complex analysis of link structure to optimize the uses of adhoc associations.

5.0 CONCLUSIONS

This dissertation views the Semantic Web as a system for the organization and discovery of information resources using classification and deduction in the Web environment. It emphasizes the values of classification and deduction to supplement the organization of information using association on the Web. What has been demonstrated is not a new approach to the organization of information. What is new is the use of an integrated system incorporating association, classification, and deduction in the finding of Web resources. This chapter makes some recommendations for approaches that could potentially lead to the construction of a "More Semantic Web".

5.1 SEMANTICS ON THE SEMATIC WEB

According to Berners-Lee, the Semantic Web will provide optional information on the Web that will facilitate machine operation. The information will be given in a well classified form with clearly defined semantics. Generally, the information on the Semantic Web will come in two forms: metadata and ontologies. RDF and the Web ontology languages were designed for the creation of metadata and ontologies for the Semantic Web. Resource semantics is a broad term which includes both metadata and ontologies

5.1.1 Metadata and Ontologies

The term *Metadata* is data used to describe resource information. Metadata provides descriptions of resources -- both information resources and non-information resources. Metadata allows

resources to be assigned to categories. Further, it allows properties of resources to be specified or associated with other resources.

The term *Ontology* is used to describe relations among categories and properties. In particular, if creating metadata is analogous to creating a catalog card for a resource, creating an ontology would be analogous to creating a subject classification. An ontology could provide for the arrangement of subject categories in a hierarchy. For example, categories could be defined similar to that of the Dewey decimal classification system, where subcategories are defined hierarchically. Further an ontology could provide definitions of new categories created in terms of existing categories. In particular, some expressive power, such as logical connectives, quantifier and cardinality, can be used in creating definitions for new categories. Ontologies could also provide the arrangement of properties in hierarchical order, i.e. a property defined as a sub-property of another property. Although the notion of sub-property is not as common and straightforward as that of subcategory, one can see some particular uses. For example, the "published-by" and "distributed-by" properties could be considered sub-properties of the "available-from" properties. This implies that resources are usually available from those who publish or distribute. Ontologies could provide the definitions of properties used in terms of links. Some expressive power, such as domain, range, inverse, symmetry and transitivity, can be used in creating definitions for such properties.

5.1.2 Associative and Classificatory Semantics

Associative links may also be included as *Resource Semantics*. Where classification is often depicted by tree structure, association is often depicted by graph structure,. In a way, they are closely related, i.e. a tree is a graph that is acyclic. Put more formally, classification implies hierarchical structure where association allows more random structure. This separation criterion is based solely on structure regardless of the underlying semantics.

From a less formal viewpoint, classification often deals with properties of groups, where association often deals with properties of "individual items". In the context of the Semantic Web, metadata created using RDF will associate individual resource with other entities, i.e. categories

and other resources. Such relationships could be denoted by an RDF graph. Thus, the notion of metadata often resembles associative semantics. Ontology, in contrast, usually involves properties of classes and properties. In addition, the relationships created in an ontology are normally hierarchical[1]. Thus, the notion of ontology often resembles classificatory semantics.

5.2 THE ROLE OF DEDUCTION IN THE SEMANTIC WEB

Deduction can supplement the organization and discovery of information resources on the Semantic Web in three ways. First, it will allow for semantics. Second, it will provide more effective information retrieval. Third, it will allow for more efficiency in the organization of information. These can be described in more detail as follows.

5.2.1 Deduction as a Means for Semantic Information Retrieval

Information retrieval on the Web is mostly based on full-text indexing. There are some limitations associated with full-text indexing -- in particular, the search terms used in the queries must match with the words appearing in the documents. Although stemming, latent semantic indexing, and clustering endeavor to overcome this requirement, these systems still rely largely on the selections of words used in the queries and in the documents. Further, the expression of user queries is limited by syntactic representation. In particular, there is no efficient mechanism in full-text indexing that allows users to describe information needs precisely and meaningfully.

One approach to addressing the semantics of documents is the use of metadata. Metadata does not rely on words in documents. It allows for a more precise and meaningful search based on keyword matching on particular attributes. Although metadata search does not rely on the words contained in documents, it still relies on word used in describing them. Thus, in a way, metadata is susceptible to the same limitations as full-text indexing.

[1] The Web Ontology Language (OWL) specification allows cyclic in category definition. Thus, there are some rare cases where the hierarchical assumption in category definition could be violated.

Searching based on deduction takes metadata a further step. In particular, in order to overcome the limitation of word variations, ontologies must be used in combined with metadata. It is deduction based on ontology that will allow the search to be based on semantics rather than simple word matching. Specifically, a deduction system will interpret the meanings of words based on an ontology. If the semantics of the query matched with the semantics describing the resources, the resources would be retrieved. A model for the retrieval of information resources using deduction system is shown in Figure 5.1.

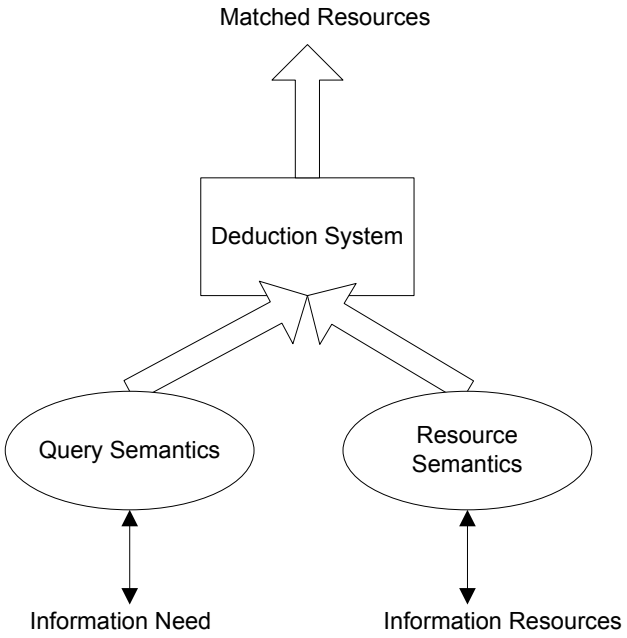


Figure 5.1. Retrieval of information resources using deduction system

With deduction enabled, query semantics do not need to be described the same way the relevant resources are described in metadata. The only requirement is that they have the same implication based on the definitions defined in the related ontologies. Put another way, a deduction system will retrieve the relevant resources whose descriptions are explicitly or implicitly matched with the query semantics. For example, a resource could be described as having as its subject "*Ronald Reagan"*. If an ontology on the US presidents specifies that, "*Ronald Reagan"* is a "*US president"*, a query for *information resources on the US presidents*

would result in the retrieval of the resource on Ronald Regan, even though the query description is not directly matched with the resource description.

Information retrieval based on semantic matching could be viewed as an alternate method to information retrieval based on keyword matching in full-text search. Although both rely on different forms of information and processing, they could complement each other in helping users in locating information.

5.2.2 Deduction as a Means for Effective Information Retrieval

By design, information resources on the Web are organized in a decentralized and ad hoc fashion. The Semantic Web proposes to provide a better organization of resources. In particular, information resources will be classified based on categories and properties as they are posted. Ideally, if resources are classified perfectly, it would allow users to find relevant resources to every information need. However, classification can be imperfect. Specifically, any given classification system responds well to queries that match the classification structure and poorly to queries that do not. The precision of classification system is maximized only when information need is closely matched to the provided categories.

When the query is more specific than the provided categories, precision of the retrieved resources will be degraded. Deduction could be applied to provide better precision in such cases. In particular, deduction could be made based on existing classification to provide better accuracy to the more specific needs. Thus, deduction will supplement classification of resources on the Semantic Web by allowing for more precise searches.

Finally, deduction could help to provide a more comprehensive association system by revealing some implicit associations between the resources. This could lead to the discovery of the relationships that were previously omitted and undiscovered. This is closely related to the discussion in the following section.

5.2.3 Deduction as a Means for Efficient Information Storage

From the viewpoint of information organization, deduction could allow for more efficient system of information about resources. In particular, some information may be omitted if it can be inferred by the deduction system. The amount of information omitted could be measured by the number of facts that need not be stated. In the context of the Semantic Web, omissions are permitted both in classificatory and associative semantics when deduction is applied. The follows provide some examples.

Subsumption relationships that could be deductively determined need not be explicitly stated. More generally, categories described in terms of existing categories imply subsumption relationships. Such implied relationships are allowed to be omitted in creating classificatory semantics. If a subject category "US history in 19th century" is to be defined in terms of two existing subject categories: "US history" and "19th century history". One definition could be that the category "US history in 19th century" is a subcategory of the category "US history" as well as a subcategory of the category "19th century history". Alternatively, it could also be defined that the category is equivalent to the conjunction of the categories "US history" and "19th century history". Because one definition implies the other, only one definition is required and the other may be omitted.

Accordingly, resources that are members of the categories that are related based on subsumption could be partially omitted. Given the above example, resources could either be explicitly stated as members of the category "US history in 19th century" or members of both individual categories "US history" and "19th century history". Both forms have the same implication based on subsumption. Thus, either form could be used while the other form is allowed to be omitted.

To provide some rough indications of the amount of information that could be omitted in creating associative semantics, some use cases are exemplified as follows. When an association type is *transitive* and there are *n* resources connected consecutively by it, the information about the resource relationships required could be reduced from the magnitude of *(n-1) n/2* to *n-1*.

When an association type has an inverse and there are *n* resources connected consecutively by it, the information about the resource relationships required could be reduced from the magnitude of *2(n-1)* to *n-1*. When an association type could be arranged in hierarchical order with *m* parent levels above it and there are *n* resources connected consecutively by it, the information about the resource relationships required to be stated could be reduced from the magnitude of (*m+1) (n-1)* to *n-1*.

Consider a case when there are ten successive versions of resources published. In order to state their relationships in terms of "newer version" and "older version", without deduction, there would be 90 statements about such relationships. With deduction applied, the "newer version" and "older version" could be defined as inverses of each other and each type could be defined as transitive. Thus, the number of relationships required could be reduced to nine statements. In addition, when a newer version of resource is published, without deduction, 20 statements about the new relationships would be required. With deduction, only one statement would be required. Thus, one can organize information with less effort when deduction is involved.

5.3 A SIMPLIFIED ARCHITECTURE

A simplified architecture for the Semantic Web data and systems is provided in Figure 5.2. Similar in operation to search engines on the Web, the deployment of the deduction system on the Semantic Web involves three tiers -- the user tier, the information tier and the deduction system tier. In this particular framework, the information tier consists of resource collections and added-semantics providers. The deduction system tier is the semantic processing unit. The user tier posts the query semantics and manages the results from the deduction system. The deduction system acquires the information and knowledge from collections and semantics providers and processes them in a decentralized fashion.

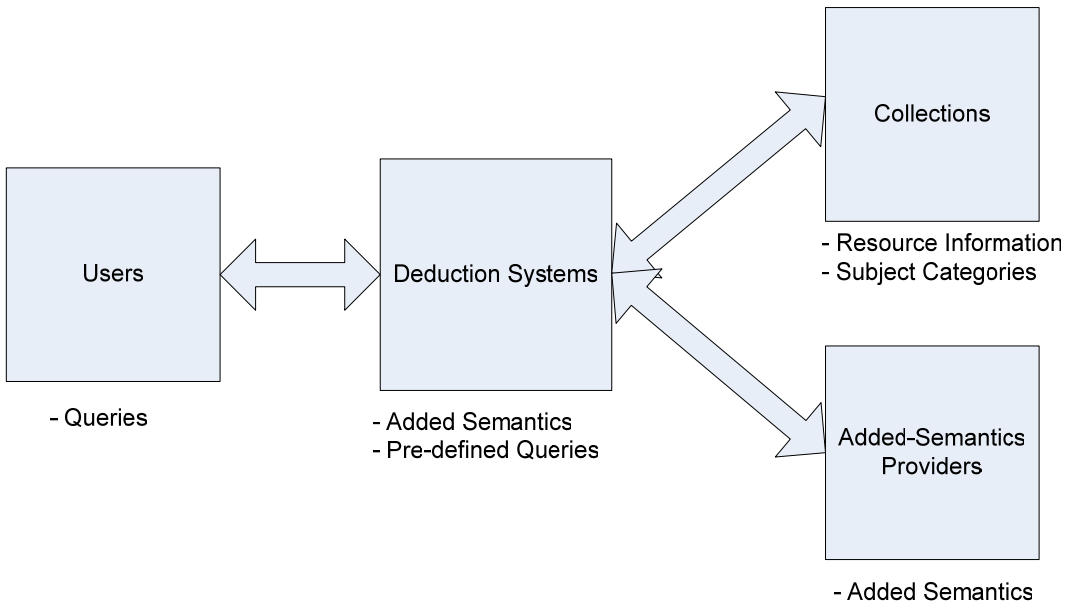


Figure 5.2. Recommendation on deployment architecture

Collections provide metadata on the information resources. Subject categories could be additionally provided by collections. Metadata is assumed to be provided in an RDF format. Ideally, via RDF, the data from multiple collections could be processed by the deduction system as a virtual homogeneous source. However, in reality, this could be difficult to achieve because of inconsistency or conflicts in the information obtained across collections. For example, the publication year of a resource could be given differently by different collections. In such a case, the deduction system must resolve inconsistencies. This architecture makes no assumption about inconsistency resolution schemes used by a deduction system.

Added-semantics allow the deduction system to make additional inferences. For example, the semantics on the US presidents and Seafood Cookery used by the case study exemplify added semantics. Added semantics are likely to be created using RDF and Web ontology languages. Added-semantics providers are presumed to be authorities on the subjects. It is also possible there will be multiple sources of semantics on the same subject. In such case, it is assumed that deduction systems and users will make the decision on which semantic providers will be chosen for particular subjects. It should be noted that the deduction system could also act as an added-semantics provider by adding its own semantics.

Deduction systems will acquire resource semantics via HTTP. The list of collections that are targets for processing could be pre-defined. The referenced semantics will be followed based on URI referencing mechanisms. The processing of resource semantics will occur in a decentralized fashion, i.e. the information processing will be independent of the information sources. Although the implementation of a deduction system based on description logic has been used, the architecture is consistent with any system supporting similar operations. The deduction system must provide for the retrieval of the processed information and knowledge through some software interface.

Finally, there must be an interface between users and the deduction system. Such a tool must provide an interface that allows the composing and posting of user queries to the deduction system as well as the processing of the returned results. This architecture makes no assumption about the design of such tool. Further, it makes no assumption about the standards in composing and posting the queries, e.g. query languages and syntax.

5.4 SOME GUIDELINES FOR DEVELOPING SEMANTICS

This section provides some guidelines in creating effective semantics for the Semantic Web. The guidelines were developed based on observations and lessons learned from the case study. Effective semantics are keys to facilitating the effectiveness of deduction systems.

Generally, deduction capability relies on information and knowledge. In the context of the Semantic Web, information and knowledge will be made available as metadata and ontologies. Deduction relies not only on the quantity but also the quality of the information and knowledge provided. Put another way, while adding information and knowledge will lead to more conclusions, inaccurate information and knowledge will lead to inaccurate conclusions. Thus, the key to the effectiveness of a deduction system is providing sufficient information and knowledge while minimizing inaccuracies.

5.4.1 Provide Sufficient Semantics

Multiple classifications facilitate deduction. In particular, classificatory semantics across many dimensions will facilitate deductions. Resources that are classified based on many different properties -- authorship, publication year, format, subject, will provide a better basis for deduction than categories classified by just a single dimension. Using another example, subject categories on Art History classified based on artists, periods, styles and regions will promote better deduction than just one dimension.

Classificatory semantics with finer granularity often provide a better basis for deduction. Structurally, this implies classificatory semantics that promotes greater hierarchical depth. For example, defining the subject category for the Java programming language in terms of "Programming Languages/ Object-Oriented Languages/ Java" is recommended rather than "Programming Languages/ Java". Using the first form, the deduction system can relate the programming language Java as an object-oriented programming language. Such deduction would not be possible using the latter form and thus deduction capability will be more limited. Further, extending hierarchical levels to classificatory semantics could significantly promote deduction capability. For example, by extending classificatory semantics for the "Seafood Cookery" subject topic, as shown in the case study, it enabled additional inferences that were not possible given the previous knowledge.

Finally, the addition of associative semantics will facilitate deduction. For example, the information on the US presidents used in the case study provides a minimalist example of associative semantics. It is possible to extend such semantics to include further associations with the related individuals, such as the first ladies, the vice presidents and other cabinet staff. This would provide a deduction system with extended knowledge that would allow further deduction on the subject.

5.4.2 Minimize Errors and Omissions

Increased information and knowledge often carries with it increases in errors and omissions. The case study has suggested that inaccuracies could increase with the amount of information provided. Errors and omissions due to misrepresentation of semantics are dangerous for a deduction system. The case study has shown that such errors and omissions could have some major impacts on the retrieval performance of a deduction system. This section provides some guidelines, based on the results of the case study, for creating metadata, ontologies and queries that avoid such pitfalls.

5.4.2.1 Minimize Inaccuracies in Metadata The three simple rules for effective metadata are *accuracy*, *specificity* and *completeness*.

The information about resources must be accurate. It was observed from the case study that one common mistake in resource cataloguing is relying on keywords appearing in a resource title. Although cataloging based on title is effective for many resources, sometimes ambiguity in word meanings could lead to cataloging errors. Such errors resemble those made by full-text indexing engines. Thus, care needs to be taken to catalogue resources based the semantics rather than the words used in representing them. Inaccurate classification will eventually lead to inaccurate results.

The information about resources should be as specific as possible. Specificity of resource information will help to promote deduction. The case study has shown that using general terms to describe resources reduces the likelihood of resources being retrieved. For example, a resource having the subject topic on *Salmon cookery* should be cataloged as such rather than as *seafood cookery*, which is a more general topic. While the more general information could often be inferred based on the more specific information, the more specific information generally could not be inferred based on the more general information.

Finally, the information about resources should be complete. In particular, resources should be described in as much detail as possible without omission. Ideally, this implies that resource should be assigned to every category which is applicable. Further, it implies that resources should be associated with every related resource. The case study has shown that the omissions in such aspects were the major causes of resources not being retrieved. Although ideally omissions must be minimized, practically this will require a tremendous effort. This is especially true in a large and volatile environment such as that envisioned by the Semantic Web.

5.4.2.2 Minimize Inaccuracies in Ontologies The case study has shown that many inaccuracies in the retrieval of resources were caused by inaccuracies in the category hierarchy.

One of the most effective ways to minimize inaccuracies in category hierarchy is to maintain subsumption integrity. In particular, a deduction system relies heavily on subsumption relationships between categories. Thus, maintaining subsumption integrity will result in more accurate conclusions. Practically, this implies that subcategories should be assigned based on subsumption. A rule of thumb would be to check whether all the resources of the assigning subcategory are also applicable to its parent category. If not, one should consider removing it as a subcategory.

In many cases, subcategories could be assigned based on the *part-of* relationships. For example, subject categories on the cities could be assigned subcategories of the countries the cities are parts of. Strictly, the part-of relationship is not semantically equivalent to subsumption. However, practically, they are sometimes indistinguishable and used interchangeably. The case study did not investigate whether subcategories assigned based on part-of relationships and subsumption will have different impacts on retrieval accuracy. Such a form of subcategory assignment should be used with caution.

Classification system should provide clear distinctions between the general and specific semantics of the categories. To represent the general semantics for a category, one could create a subcategory for it. For example, the category “General” can be created as a subcategory of the category “Programming Language” for the resources on the general aspect of the subject. Thus,

when deduction is required based only on the general aspect, the category "Programming Language/ General" can be processed by deduction system. When deduction is also required based on the specific aspect, the category "Programming Language", whose semantics will include all of its subcategories, can be used to imply both the general and the specific aspects of the subject.

Properties should be defined with accuracy and specificity. In particular, subsumption integrity in property hierarchy should be maintained. Inverse relations should be defined to allow the discovery of omitted information in a bi-directional relationship. Transitivity should be applied where necessary to allow better discovery of omitted information. However, transitivity and transitivity with inverse should be defined with caution. Specifically, inaccuracies are sensitive to these forms and could propagate rapidly by them.

5.4.2.3 Minimize Inaccuracies in Queries Inaccuracies and omissions in composing queries could have the most severe impacts on the retrieval performance. The case study has shown that retrieval performance was significantly degraded when queries were misrepresented. It was observed that several factors could contribute to inaccuracies and insufficiencies in queries. One relates to the semantics of the category terms. Another relates to the expressive power utilized in query expression.

The first form of inaccuracy is often caused by the mismatch between the semantics of the category terms implied by the queries and those implied by the collection. For example, a query expression could refer to the category term "Cooking/ Dessert" to imply the resources on cakes and pies, however, if such a category in the collection also implies cookies and ice-cream, the retrieved results would be inaccurate. In order to alleviate such problems, collections should provide clear category descriptions, i.e. what is implied and not implied by each category. In addition, it must be clear whether the meaning in the general or specific sense of the category is being referred to. In particular, query must be specified clearly whether it refers to the broad or specific sense of the category as suggested in section 5.4.2.2.

Omission in composing queries is often caused by users' lack of knowledge about the existing vocabularies. For example, users may fail to include some category and property terms because of ignorance. One solution is a tool that will allow user to search for related vocabularies, i.e. category and property terms. Such tools could be based on hierarchical navigation or search based on descriptions of categories and properties. Alternatively, some forms of pre-defined queries could be created to allow users to use queries without knowledge in composing them. Pre-defined queries could be created by the deduction system or collection. Descriptions of the pre-defined queries should be clearly provided to ensure that users can select the queries that match with their needs.

Another issue in composing queries relates to the use of expressive power. The case study has shown that some uses of expressive power could impact the retrieval performance. In particular, the results of the queries based on negation can be volatile. Thus, users should be aware that, although negation can provide efficiency in expressing information need, it is susceptible to inaccuracies and should be used with cautions. In some cases, users may consider using disjunctions or quantifiers as alternatives to using negation. For example, instead of expressing information need based on negation, such as resources on "*non-alcoholic beverages*", an alternative expression based on disjunction, such as resources on "*tea or coffee or juice*", should be considered. Further, the case study has shown that verbose query expression could sometimes help in improving retrieval performance. Query semantics that are composed accurately, sufficiently and expressively are likely to result in a good retrieval performance.

Finally, the case study has provided some evidence of individual differences in interpretation of semantics. In particular, there was a degree of disagreement among the human judges on particular subjects. Similar disagreement could also arise among users and could impact retrieval performance. The composition of queries should incorporate the choices of semantics from variety of semantics providers. This will allow users to choose the semantics that are consistent with their perspective. This could help in reducing the impacts of individual differences in semantic interpretation on the retrieval performance.

5.5 DISCUSSION OF IMPLEMENTATION ISSUES

This dissertation used one implementation based on RDF, DAML and a description logic system. This section discusses some issues and limitations related to the implementation. In particular, it focuses on some deviations from the standards and some limitations that future researchers should be aware of in researching similar systems.

5.5.1 Implementation of RDF and Ontology Language

A major deviation in the implementation from the RDF standard is in the graphical notation. It was found that the RDF graph notation was not clear enough to illustrate some key elements that the implementation is based on. In particular, the implementation requires the clear distinction between class, relation, and instance. However, using the RDF graph notation, it is often difficult to visually differentiate them, i.e. they are indistinguishably represented as oval shapes. The implementation uses the ad hoc notations to improve clarity. In particular, it uses different graphical shapes in representing class, relation and instance. Further, it separates the modeling of class hierarchy, relational hierarchy and resource associations. This helps to provide clarity in the modeling of classificatory and associative semantics. Although the non-standardized notations were used, they can be straightforwardly serialized into RDF metadata and ontologies.

Ontology processing, discussed in section 3.3.4, was based on the DAML+OIL language. As of February 2004, the DAML+OIL language has been superseded by OWL, which is the current standard for Web ontology language. Although the implementation was created based on DAML+OIL, some adjustments would allow it to be applied to OWL. For example, the mapping between the expressive power of the ontology language to that of description logic can be achieved similarly in DAML+OIL and OWL-DL.

5.5.2 Implementation of the Description Logic System

Several limitations of the description logic system were found. One relates to the lack of support in providing retrieval service based on the closed-world assumption. The case study exemplified

some circumstances where the closed-world assumption would be required. However, description logic system which used the open-world assumption will not produce useful results in such circumstances. Even though some preprocessing techniques were used to achieve the desirable outcomes, they will not scale well.

There were several cases where representations were found limited by existing expressive power. For example, *autobiography* book is a book where the person in the subject is identical to the person in the author. Although the expressive power permits the expression, where the person in the subject has identical property as the person in the author, e.g. *books on the US presidents authored by the US presidents*, it was impossible to create the semantics for *autobiography books by the US presidents*. A similar insufficiency in expressing identical instances could be exemplified in another example. Based on the associative semantics defined for the US presidents (Figure A.4), it was impossible to prevent the conclusions that the former US president *Cleveland Grover* (1885-1889, 1893-1897) is a US president preceding and succeeding himself. Thus, it should be aware that permitting expressive power could limit the sufficiency and accuracy in representing semantics.

Finally, a deduction system for the Semantic Web must deal with a large volume of information. If a description logic system is to be used for such purpose, it must have good computational efficiency. In particular, it must respond in a timely fashion given the variable amount of processing information and knowledge. However, it is as yet unknown whether any description logic system will be able to perform efficiently. Computational efficiency of the deduction system is beyond the scope of this dissertation.

5.6 THE FUTURE OF THE SEMANTIC WEB

It should be emphasized that the results from the case study were obtained in a simulated setting simplifying those of the Semantic Web. One of the simplifications was the use of the data from a single resource collection to simulate the data on the Semantic Web. In addition, the case study incorporated added semantics that were relatively simple. To fully realize the potential of the

Semantic Web, the deduction system must combine the information and knowledge obtained from multiple collections and semantics providers. Further, resource semantics provided will be complex. A deduction system in a more complex setting must deal with various levels of complexity that have been simplified in the case study.

One of the major issues is the consistency of the resource semantics obtained from multiple sources. In particular, when resource semantics are obtained from different sources, consistency can not be guaranteed. Information provided by one source could be inconsistent with another source. Deduction systems must be able to produce reliable conclusions under such conditions. Although currently there is no standardized approach in maintaining consistency of the Semantic Web, some general approaches have been envisioned by Berners-Lee, based on proof and trust (Figure 1.2). Generally, such mechanisms are likely to provide some verification and non-repudiation mechanisms for the information and knowledge obtained. Progress in such topics will be critical to the scalable deployment of the framework.

In conclusion, this dissertation has elaborated a framework based on some fundamental principles of the Semantic Web related to the organization and discovery of information resources. The potential of the Semantic Web in such aspects have been explored in an integrated fashion using a large data set and applying objective metrics from the field of information retrieval. It must be emphasized that the ultimate scope of the Semantic Web is still beyond that elaborated in this dissertation research. It has been suggested that some applications could be benefit from the Semantic Web, e.g. Agents, Web Services, Expert systems, Decision support systems, etc. Further, the deployment of scalable Semantic Web applications will involve many challenging problems such as ontology integration, ontology maintenance, reasoning under uncertainty, information privacy and security, etc.

The success of the Semantic Web will also largely depend on users' awareness of its potential. Such awareness, together with the developing standards and technologies, will accelerate the development of the Semantic Web. Further, like the Web, it is speculated that individual contributions will increase the value of the Semantic Web. One goal of this research has been to elevate the awareness of the potential of the Semantic Web using framework and

methodology of information sciences. Further understanding will be required and disseminated toward the realization of the Semantic Web.

APPENDIX A

ADDED SEMANTICS

A.1 ADDED CLASSIFICATORY SEMANTICS FOR MEDIA FORMATS

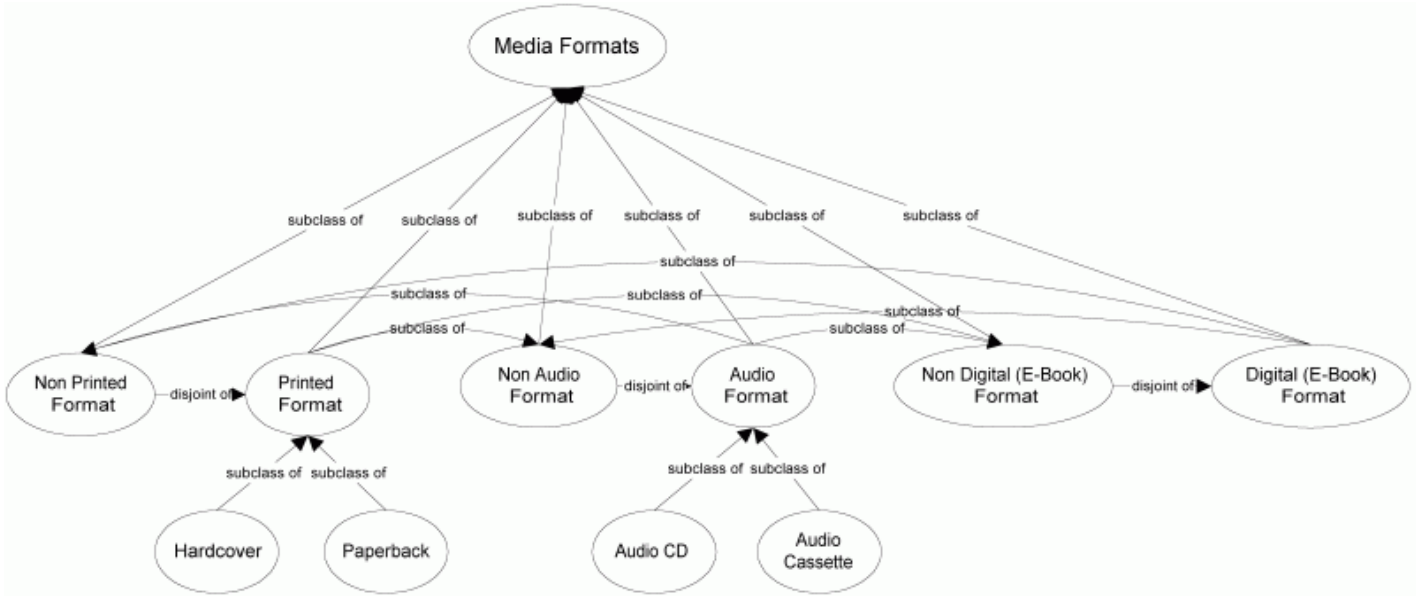


Figure A.1 Added Classificatory Semantics for the Media Format Classes

A.2 ADDED CLASSIFICATORY SEMANTICS FOR SEAFOOD COOKERY TOPIC CLASSES

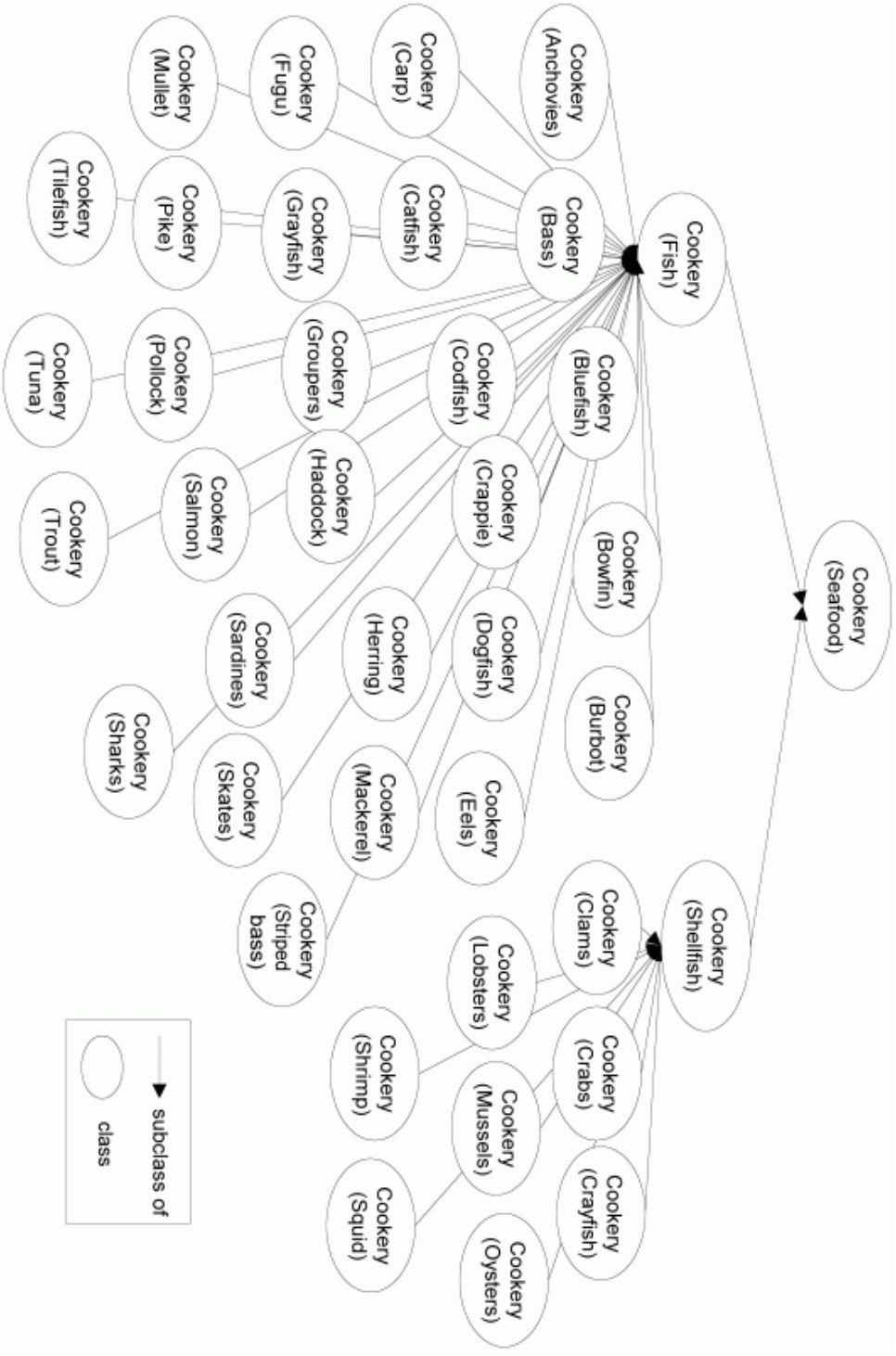


Figure A.2 Added Classificatory Semantics for the Seafood Cookery Topic Classes

A.3 ADDED CLASSIFICATORY AND ASSOCIATIVE SEMANTICS FOR THE US PRESIDENT TOPIC CLASSES

A.3.1 Added Classificatory Semantics

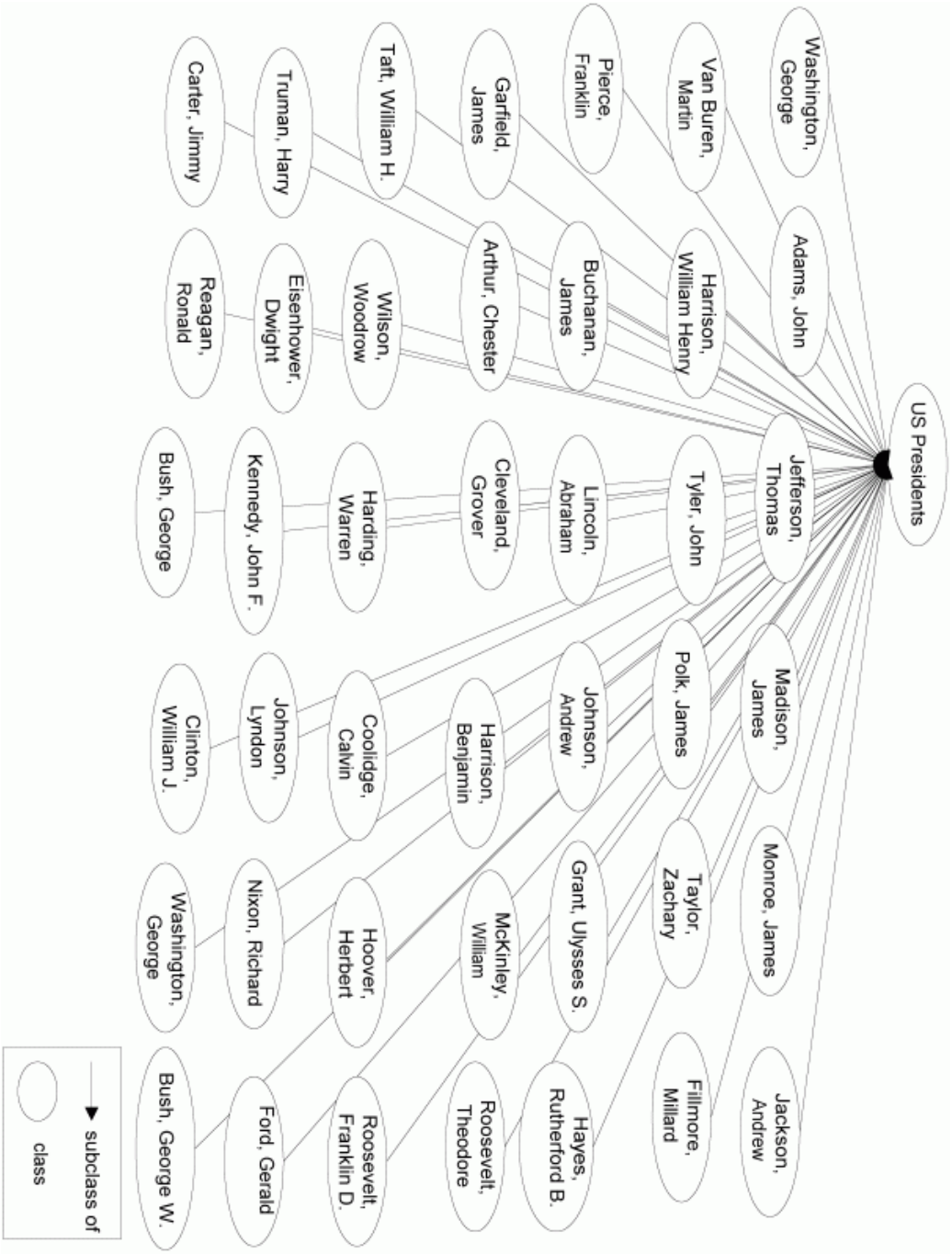


Figure A.3 Added Classificatory Semantics for the US Presidents Topic Classes

A.3.2 Added Associative Semantics

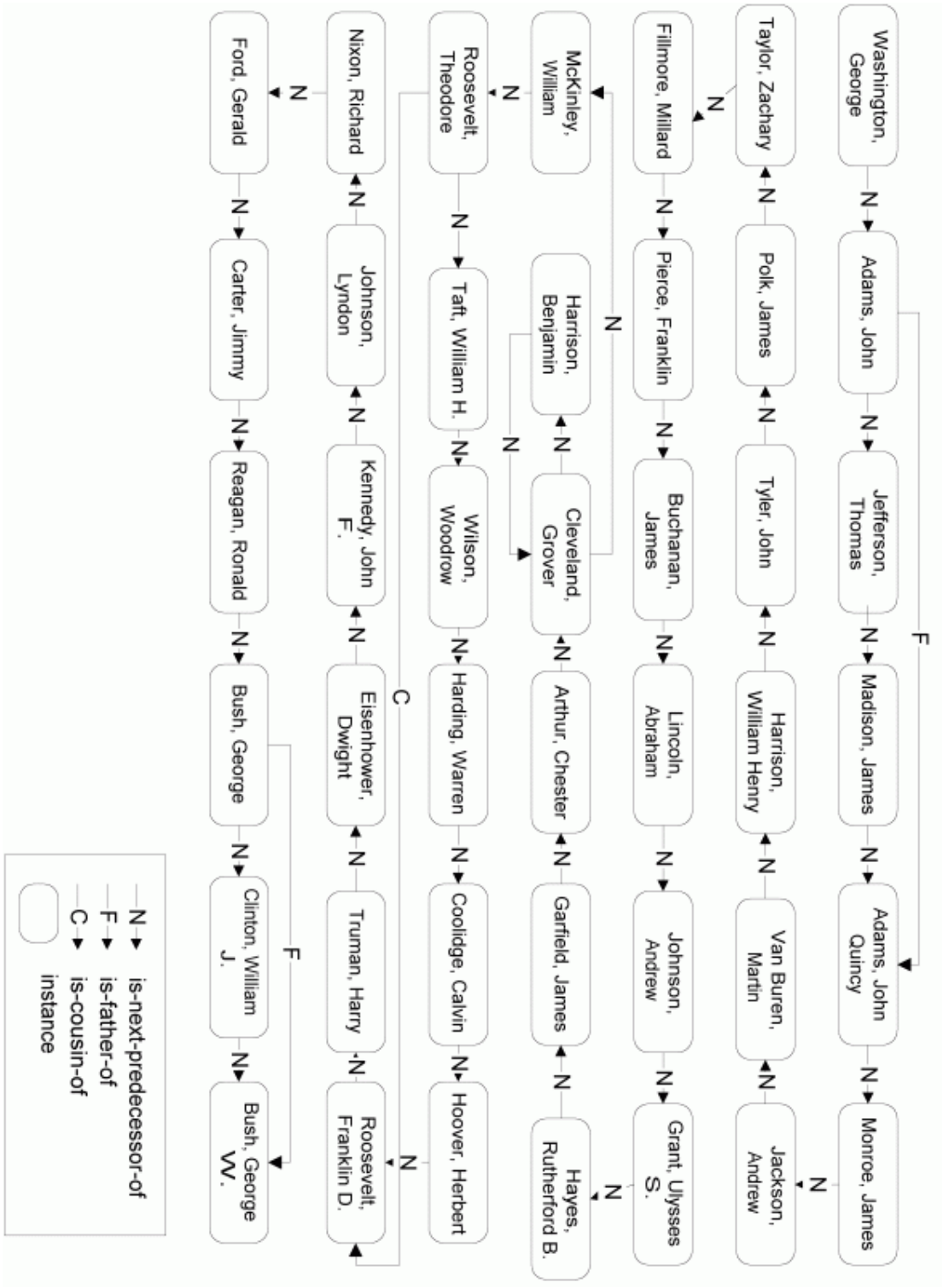


Figure A.4 Added Associative Semantics for the US Presidents Topic Instances

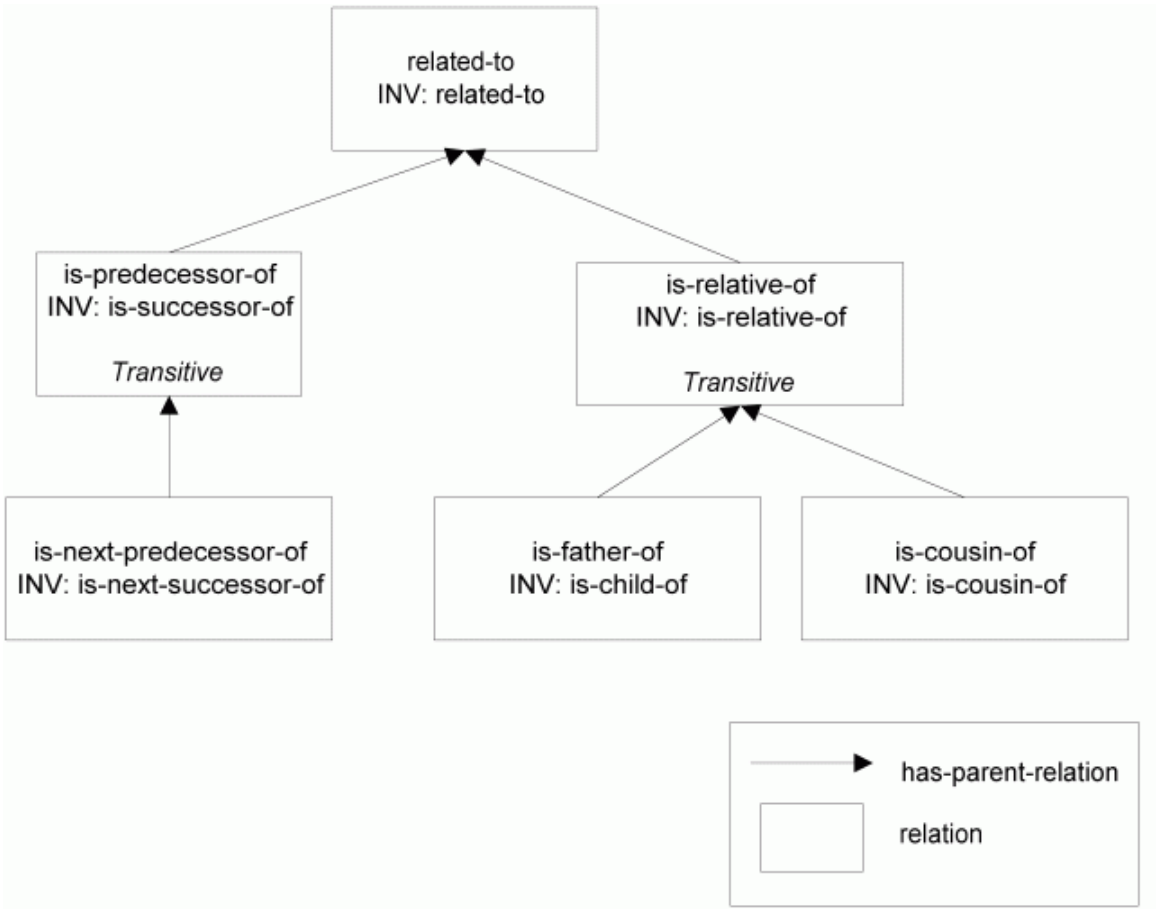


Figure A.5 Relation Definitions for the US Presidents Topic Instances

APPENDIX B

QUERIES FOR THE ANALYSIS OF RETRIEVAL EFFECTIVENESS

The descriptions of the queries are provided in Table B.1. The translations of the queries into query expressions are provided in Table B.2. The RACER DL syntax is used for the expressions. The information on the subject categories is provided in Table B.3. The additional defined classes that are used by the queries are represented using "CD" prefix. Their definitions are provided in Table B.4.

Table B.1. Descriptions of the queries for the analysis of retrieval effectiveness

QID	Query Descriptions
1	Books on Database technology from the publisher O'reilly
2	Books on Database technology from the publisher O'reilly, except those on Oracle database system
3	Books on biography of Japanese or Chinese women
4	Books on the history of Iraq in relation to Iran (or vice versa)
5	Books on biography of world leaders in the 19th century
6	Books on French dessert or French pastry baking
7	Books on Italian cooking except those on pasta
8	Books on meat cooking (non-seafood)
9	Books on non-alcoholic beverages

QID	Query Descriptions
10	Books on Chinese or Japanese vegetarian cooking
11	Books on non-seafood Chinese or Japanese cooking
12	Books on traveling in the Benelux region (Belgium, Netherlands and Luxemburg), except those on Amsterdam or Brussels
13	Books on traveling in the Benelux region, which contain the information of the three countries in a single book
14	Books on traveling in the Benelux region, which each book contain the information of each individual country
15	Books on traveling in the Benelux region (Belgium, Netherlands and Luxemburg) from the Lonely Planet or Eyewitness guidebook series
16	Books on the series Star Wars written by George Lucas
17	Books written by George Lucas except those on the series Star Wars
18	Books on database programming using Java
19	Books on security in e-commerce
20	Fiction books on war or sea adventure except those written by Ernest Hemingway
21	Fiction books written by Ernest Hemingway except those on war or sea adventure
22	Maps of the Pittsburgh areas
23	Books on traveling in South East Asia from the Eyewitness or Lonely Planet guidebook series
24	Maps of the South East Asia countries
25	Books on history of piano
26	Books on Dutch art history
27	Books on European art history except those on Dutch art
28	Books on non-European art history
29	Books on the history of Asian paintings

QID	Query Descriptions
30	Books on the history of paintings in the United States
31	Books on international (non-American) art history
32	Books on Microsoft Office 2000 or Office XP in the for-dummies book series
33	Books co-authored by Arthur C. Clarke
34	Books solely authored by Arthur C. Clarke (single author)
35	Books co-authored by Arthur C. Clarke and Gentry Lee with or without others
36	Books co-authored by Arthur C. Clarke and Gentry Lee only (no others)
37	Books by Arthur C. Clarke published in 1970s
38	Books by Arthur C. Clarke published in 1960s or 1970s
39	Books by Arthur C. Clarke published between 1960-1965 or between1970-1974
40	Books on war fictions in printed format (hardcover or paperback)
41	Books on war fictions in audiobook format (cassette or CD)
42	Books on war fictions in non-audiobook format
43	Books on war fictions not in audiobook or e-book format
44	Books on Java Programming published by the publisher of the book "Java How to Program, Fifth Edition" (ISBN: 0131016210)
45	Books on Java Programming solely authored by one of or all the authors of the book "Java How to Program, Fifth Edition"
46	Books on Java Programming authored or co-authored by one of or all the authors of the book "Java How to Program, Fifth Edition"
47	Books on Java Programming published in the same year as the book "Java How to Program, Fifth Edition"
48	Books on fish cooking
49	Books on shellfish cooking
50	Books on either fish or shellfish cooking or both

QID	Query Descriptions
51	Books on both fish and shellfish cooking in a single book
52	Books on salmon cooking
53	Books on crabs or shrimp or lobsters cooking
54	Books on biography of the presidents of the United States
55	Books on biography of the first president of the United States
56	Books on biography of the presidents of the United States succeeding the former president John F. Kennedy
57	Books on biography of the presidents of the United States succeeding the former president John F. Kennedy but preceding the former president Ronald Reagan
58	Books on biography of the presidents of the United States preceding the former president Thomas Jefferson
59	Books on biography of the presidents of the United States who are fathers of other US presidents
60	Books on biography of the presidents of the United States who are sons of other US presidents
61	Books on biography of the presidents of the United States who are cousins of other US presidents
62	Books on biography of the presidents of the United States who are relatives of other US presidents
63	Books on biography of the US presidents authored by the US presidents
64	Books on French dessert or French pastry baking
65	Books on French dessert or French pastry baking
66	Books on fish cooking
67	Books on fish cooking
68	Books on shellfish cooking
69	Books on shellfish cooking

QID	Query Descriptions
70	Books on either fish or shellfish cooking or both
71	Books on either fish or shellfish cooking or both
72	Books on biography of the presidents of the United States
73	Books on biography of the presidents of the United States
74	Books on biography of the first president of the United States
75	Books on biography of the first president of the United States

Table B.2. Query expressions in description logic syntax

QID	Query Expressions
1	(AND C69860 C549646)
2	(AND C69860 (AND C549646 (NOT C4092)))
3	(AND C2445 (OR C2372 C2368))
4	(AND C5000 C4999)
5	(AND C4854 C2418)
6	(AND C4280 (OR C4201 C4204))
7	(AND C4285 (NOT C4217))
8	(AND C4212 (NOT C4216))
9	(AND C4219 (AND (NOT C4221) (AND (NOT C4220) (AND (NOT C4224) (NOT C4223)))))
10	(AND C4336 (OR C4266 C4269))
11	(AND (OR C4266 C4269) (NOT C4216))
12	(AND (OR C16988 (OR C16925 C16982)) (AND (NOT C67669) (NOT C67575)))
13	(AND C16988 (AND C16925 C16982))
14	(AND (OR C16988 (OR C16925 C16982)) (AND (OR (NOT C16988) (NOT C16925)) (AND (OR (NOT C16988) (NOT C16982)) (OR (NOT C16925) (NOT C16982)))))
15	(AND (OR C17101 C17078) (OR C16988 (OR C16925 C16982)))
16	(AND C15564 C281542)
17	(AND C15564 (NOT C281542))
18	(AND C549646 C3608)
19	(AND C886500 (OR C3875 C3632))
20	(AND (NOT C70323) (OR C10195 C886086))
21	(AND C70323 (AND (NOT C10195) (NOT C886086)))

QID	Query Expressions
22	(AND C11455 C67529)
23	(AND (OR C17101 C17078) (OR C16841 (OR C16803 (OR C16795 (OR C16826 C16813 (OR C16783 (OR C16849 (OR C16821 C16799))))))))
24	(AND C11453 (OR C16841 (OR C16803 (OR C16795 (OR C16826 C16813 (OR C16783 (OR C16849 (OR C16821 C16799)))))))
25	(AND C4511 C1769)
26	(AND C1100 C4968)
27	(AND C1100 (NOT C4968))
28	(AND C1095 (NOT C1100))
29	(AND C1099 C1876)
30	(AND C1876 (OR C1098 (OR C1103 C1097)))
31	(AND C1095 (AND (NOT C1098) (AND (NOT C1103) (NOT C1097))))
32	(AND (OR C4122 C4123) (OR C173199 C746142))
33	(AND C14933 (AT-LEAST 2 has-author))
34	(AND C14933 (AND (AT-LEAST 1 has-author) (AT-MOST 1 has-author)))
35	(AND (OR C14933 C15524) (AT-LEAST 2 has-author CDA1))
36	(AND (OR C14933 C15524) (AND (AT-LEAST 2 has-author CDA1) (AT-MOST 2 has-author CDA1)))
37	(AND C14933 (SOME has-publication-year CDY_YEARS1970S))
38	(AND C14933 (SOME has-publication-year (OR CDY_YEARS1970S CDY_YEARS1960S)))
39	(AND C14933 (SOME has-publication-year (OR CDY_YEARS1970_1974 CDY_YEARS1960_1965)))
40	(AND C10195 (SOME has-format CDM_PRINTED_BOOKS))
41	(AND C10195 (SOME has-format CDM_AUDIO_BOOKS))

QID	Query Expressions
42	(AND C10195 (SOME has-format (NOT CDM_AUDIO_BOOKS)))
43	(AND C10195 (SOME has-format (AND (NOT CDM_AUDIO_BOOKS) (NOT CDM_E_BOOKS))))
44	(AND C3608 (SOME has-publisher (SOME publisher-of CDB1)))
45	(AND C3608 (ALL has-author (SOME author-of CDB1)))
46	(AND C3608 (SOME has-author (SOME author-of CDB1)))
47	(AND C3608 (SOME has-publication-year (SOME publication-year-of CDB1)))
48	(AND C4216 (SOME has-topic CDTS_FISH_COOKERY))
49	(AND C4216 (SOME has-topic CDTS_SHELLFISH_COOKERY))
50	(AND C4216 (SOME has-topic (OR CDTS_FISH_COOKERY CDTS_SHELLFISH_COOKERY)))
51	(AND C4216 (AND (SOME has-topic FISH_COOKERY) (SOME has-topic CDTS_SHELLFISH_COOKERY)))
52	(AND C4216 (SOME has-topic CDTS_SALMON_COOKERY))
53	(AND C4216 (SOME has-topic (OR CDTS_CRABS_COOKERY (OR CDTS_SHRIMP_COOKERY CDTS_LOBSTERS_COOKERY))))
54	(AND C2418 (SOME has-topic CDTU_US_PRESIDENTS))
55	(AND C2418 (SOME has-topic (AND CDTU_US_PRESIDENTS (AT-MOST 0 is-next-successor-of))))
56	(AND C2418 (SOME has-topic (AND CDTU_US_PRESIDENTS (SOME is-successor-of CDTU_JOHN_F_KENNEDY))))
57	(AND C2418 (SOME has-topic (AND CDTU_US_PRESIDENTS (AND (SOME is-successor-of CDTU_JOHN_F_KENNEDY) (SOME is-predecessor-of CDTU_RONALD_REAGAN)))))
58	(AND C2418 (SOME has-topic (AND CDTU_US_PRESIDENTS (SOME is-predecessor-of CDTU_THOMAS_JEFFERSON))))

QID	Query Expressions
59	(AND C2418 (SOME has-topic (AND CDTU_US_PRESIDENTS (SOME is-father-of CDTU_US_PRESIDENTS))))
60	(AND C2418 (SOME has-topic (AND CDTU_US_PRESIDENTS (SOME is-child-of CDTU_US_PRESIDENTS))))
61	(AND C2418 (SOME has-topic (AND CDTU_US_PRESIDENTS (SOME is-cousin-of CDTU_US_PRESIDENTS))))
62	(AND C2418 (SOME has-topic (AND CDTU_US_PRESIDENTS (SOME is-relative-of CDTU_US_PRESIDENTS))))
63	(AND C2418 (SOME has-author CDTU_US_PRESIDENTS))
64	(OR (AND C4280 (OR C4201 C4204)) (AT-LEAST 2 has-similar-item (AND C4280 (OR C4201 C4204))))
65	(OR (AND C4280 (OR C4201 C4204)) (AT-LEAST 3 has-similar-item (AND C4280 (OR C4201 C4204))))
66	(OR (AND C4216 (SOME has-topic CDTS_FISH_COOKERY)) (AT-LEAST 2 has-similar-item (AND C4216 (SOME has-topic CDTS_FISH_COOKERY))))
67	(OR (AND C4216 (SOME has-topic CDTS_FISH_COOKERY)) (AT-LEAST 3 has-similar-item (AND C4216 (SOME has-topic CDTS_FISH_COOKERY))))
68	(OR (AND C4216 (SOME has-topic CDTS_SHELLFISH_COOKERY)) (AT-LEAST 2 has-similar-item (AND C4216 (SOME has-topic CDTS_SHELLFISH_COOKERY))))
69	(OR (AND C4216 (SOME has-topic CDTS_SHELLFISH_COOKERY)) (AT-LEAST 3 has-similar-item (AND C4216 (SOME has-topic CDTS_SHELLFISH_COOKERY))))
70	(OR (AND C4216 (SOME has-topic (OR CDTS_FISH_COOKERY CDTS_SHELLFISH_COOKERY))) (AT-LEAST 2 has-similar-item (AND C4216 (SOME has-topic (OR CDTS_FISH_COOKERY CDTS_SHELLFISH_COOKERY)))))

QID	Query Expressions
71	(OR (AND C4216 (SOME has-topic (OR CDTS_FISH_COOKERY CDTS_SHELLFISH_COOKERY))) (AT-LEAST 3 has-similar-item (AND C4216 (SOME has-topic (OR CDTS_FISH_COOKERY CDTS_SHELLFISH_COOKERY)))))
72	(OR (AND C2418 (SOME has-topic CDTU_US_PRESIDENTS)) (AT-LEAST 2 has-similar-item (AND C2418 (SOME has-topic CDTU_US_PRESIDENTS))))
73	(OR (AND C2418 (SOME has-topic CDTU_US_PRESIDENTS)) (AT-LEAST 3 has-similar-item (AND C2418 (SOME has-topic CDTU_US_PRESIDENTS))))
74	(OR (AND C2418 (SOME has-topic (AND CDTU_US_PRESIDENTS (AT-MOST 0 is-next-successor-of)))) (AT-LEAST 2 has-similar-item (AND C2418 (SOME has-topic (AND CDTU_US_PRESIDENTS (AT-MOST 0 is-next-successor-of))))))
75	(OR (AND C2418 (SOME has-topic (AND CDTU_US_PRESIDENTS (AT-MOST 0 is-next-successor-of)))) (AT-LEAST 3 has-similar-item (AND C2418 (SOME has-topic (AND CDTU_US_PRESIDENTS (AT-MOST 0 is-next-successor-of))))))

Table B.3. Statistics of the subject categories involved in the queries

Category ID	Category Path Name	Total Members	Total Sub-categories	Max Subcategory Depth
C10195	Literature&Fiction / GenreFiction / War	1,461	0	0
C1095	Arts&Photography / Art / ArtHistory / Regional	5,129	9	1
C1097	Arts&Photography / Art / ArtHistory / Regional / AfricanAmerican	303	0	0
C1098	Arts&Photography / Art / ArtHistory / Regional / United States	1,103	0	0
C1099	Arts&Photography / Art / ArtHistory / Regional / Asian	779	0	0
C1100	Arts&Photography / Art / ArtHistory / Regional / European	1,882	0	0
C1103	Arts&Photography / Art / ArtHistory / Regional / NativeAmerican	641	0	0
C11453	Arts&Photography / Artists,A-Z	3,510	0	0
C11455	Reference / Maps / Americas	108	0	0
C14933	ScienceFiction&Fantasy / Authors,A-Z / (C) / Clark,ArthurC.	81	0	0
C15524	ScienceFiction&Fantasy / Authors,A-Z / (L) / Lee, Gentry	8	0	0
C15564	ScienceFiction&Fantasy / Authors,A-Z / (L) / Lucas,George	49	0	0
C16783	Travel / Asia / Cambodia	125	0	0
C16795	Travel / Asia / Indonesia	619	6	1
C16799	Travel / Asia / Laos	0	0	0
C16803	Travel / Asia / Malaysia&Brunei	70	0	0
C16813	Travel / Asia / Myanmar	136	0	0

Category ID	Category Path Name	Total Members	Total Sub-categories	Max Subcategory Depth
C16821	Travel / Asia / Philiphines	198	0	0
C16826	Travel / Asia / Singapore	185	0	0
C16841	Travel / Asia / Thailand	596	2	1
C16849	Travel / Asia / Vietnam	192	0	0
C16925	Travel / Europe / Belgium	487	2	1
C16982	Travel / Europe / Luxembourg	33	0	0
C16988	Travel / Europe / Netherlands	408	2	1
C17078	Travel / GuidebookSeries / Eyewitness	473	21	1
C17101	Travel / GuidebookSeries / LonelyPlanet	963	0	0
C173199	Computers&Internet / Microsoft / Applications / Office2000	1,465	38	2
C1769	Entertainment / Music / Instrument&Performers / Piano	1,585	0	0
C1876	Arts&Photography / Art / Painting	6,322	142	2
C2368	Biographies&Memoirs / Ethnic&National / Chinese	1,128	0	0
C2372	Biographies&Memoirs / Ethnic&National / Japanese	1,623	0	0
C2418	Biographies&Memoirs / Leaders&NotablePeople / Presidents&HeadsOfState	927	0	0
C2445	Biographies&Memoirs / SpecificGroups / Women	12,600	0	0
C281542	ScienceFiction&Fantasy / ScienceFiction / Series / MediaSeries / StarWars	1,022	0	0
C3608	Computer&Internet / Programing / Java	1,389	14	1
C3632	Computer&Internet / Programing / Algorithms / Encryption	190	0	0
C3875	Computer&Internet / Programing / Algorithms / Cryptography	275	0	0

Category ID	Category Path Name	Total Members	Total Sub-categories	Max Subcategory Depth
C4092	Computers & Internet / Databases / Specific Databases / Oracle	553	0	0
C4122	Computers&Internet / Software / IntroductoryGuides / ForDummies:Applications	346	0	0
C4123	Computers&Internet / Software / IntroductoryGuides / ForDummies:General	673	0	0
C4201	Cooking / Baking / Desserts	459	0	0
C4204	Cooking / Baking / Pastry	47	0	0
C4212	Cooking / ByIngredient / Meat&Poultry&Seafood	649	4	1
C4216	Cooking / ByIngredient / Meat&Poultry&Seafood / Seafood	299	0	0
C4217	Cooking / ByIngredient / Pasta	217	0	0
C4219	Cooking / Drinks&Beverage	2,736	13	2
C4220	Cooking / Drink&Beverages / Bartending	78	0	0
C4221	Cooking / Drink&Beverages / Beer	126	0	0
C4223	Cooking / Drink&Beverages / Spirits	923	0	0
C4224	Cooking / Drink&Beverages / Wine	1,922	6	1
C4266	Cooking / Regional / Asian / Chinese	250	0	0
C4269	Cooking / Regional / Asian / Japanese	121	0	0
C4280	Cooking / Regional / European / French	219	0	0
C4285	Cooking / Regional / European / Italian	372	0	0
C4336	Cooking / Vegetarian	2,395	7	1
C4511	Entertainment / Music / History&Critisism	2,941	0	0
C4854	History / Americas / USA / 19thCentury	2,588	6	1
C4968	History / Europe / Netherlands	1,052	0	0

Category ID	Category Path Name	Total Members	Total Sub-categories	Max Subcategory Depth
C4999	History / MiddleEast / Iran	1,024	0	0
C5000	History / MiddleEast / Iraq	609	0	0
C549646	Computer&Internet / Databases	6,989	43	7
C67529	Travel / UnitedStates / States / PA / Pittsburgh	34	0	0
C67575	Travel / Europe / Belgium / Brussels	74	0	0
C67669	Travel / Europe / Netherlands / Amsterdam	195	0	0
C69860	Computer&Internet / ByPublisher / O'reilly	767	39	2
C70323	Literature&Fiction / AuthorsA-Z / (H) / Hemingway,Ernest	110	6	1
C746142	Computers&Internet / Microsoft / Applications / OfficeXP	181	0	0
C886086	Literature&Fiction / GenreFiction / SeaAdventure	396	0	0
C886500	Computer&Internet / DigitalBusiness&Culture / E-commerce	781	0	0

Table B.4. Definitions of the defined classes used by the queries

Defined ClassID	Defined Class Definitions
CDA1	*{iArthur_C_Clarke, iGentry_Lee}*
CDB1	*{i0131016210}*
CDM_AUDIO_BOOKS	*See class "Audio Format" in Appendix A.1*
CDM_E_BOOKS	*See class "Digital Format" in Appendix A.1*
CDM_PRINTED_BOOKS	*See class "Printed Format" in Appendix A.1*
CDTS_CRABS_COOKERY	*See class "Cookery (Crabs)" in Appendix A.2*
CDTS_FISH_COOKERY	*See class "Cookery (Fish)" in Appendix A.2*
CDTS_LOBSTERS_COOKERY	*See class "Cookery (Lobsters)" in Appendix A.2*
CDTS_SALMON_COOKERY	*See class "Cookery (Salmon)" in Appendix A.2*

Defined ClassID	Defined Class Definitions
CDTS_SHELLFISH_COOKERY	*See class "Cookery (Shellfish)" in Appendix A.2*
CDTS_SHRIMPS_COOKERY	*See class "Cookery (Shrimp)" in Appendix A.2*
CDTU_JOHN_F_KENNEDY	*See class "Kennedy, John F." in Appendix A.3*
CDTU_RONALD_REAGAN	*See class "Reagan, Ronald" in Appendix A.3*
CDTU_THOMAS_JEFFERSON	*See class "Jefferson, Thomas" in Appendix A.3*
CDTU_US_PRESIDENTS	*See class "US Presidents" in Appendix A.3*
CDY_YEARS1960S	*{i1960, i1961, i1962, i1963, i1964, i1965, i1966, i1967, i1968, i1969}*
CDY_YEARS1960_1965	*{i1960, i1961, i1962, i1963, i1964, i1965}*
CDY_YEARS1970S	*{i1970, i1971, i1972, i1973, i1974, i1975, i1976, i1977, i1978, i1979}*
CDY_YEARS1970_1974	*{i1970, i1971, i1972, i1973, i1974}*

APPENDIX C

RESULT REPORTS

Table C.1 reports the proportion of relevant resources found in the review sets. It shows the number of resources reviewed for each set, the proportion of relevant resources found in the result set (p_d), the proportion of those found in the control set (p_c) and the proportion of non-retrieved relevant resources found (p_{cnr}). The estimated proportions are provided along with the lower (L) and upper (U) limit of the 95% confidence interval.

Table C.2 reports the information on the control sets, i.e. the number of resources in the control set (N_c), the number of relevant resources in the control set, which is measured based on the proportion of relevant resources found and the control set size (p_cN_c) and the number of non-retrieved relevant resources, which is measured based on the proportion of non-retrieved relevant resources found and the control set size ($p_{cnr}N_c$). The control set precision is measured by the proportion of relevant resources found in the control set (p_c). F-measure is measured by *2*Precision*Recall/(Precision+Recall)*. Thus, F-measure of a control set is computed as *2*Precision/(Precision+1)*. Deduction impact index is the difference between F-measure of result set and F-measure of control set for a query.

Table C.3 reports the information on the result sets, i.e. the number of resources in the result set (N_d), the number of relevant resources in the result set, which is measured based on the proportion of relevant resources found and the result set size (p_dN_d), the precision of the result set, which is measured by p_d, and the recall of the result set measured using two methods

($Recall_1$ and $Recall_2$). $Recall_1$ is measured based on p_d and p_c, while $Recall_2$ is measured based on p_d and p_{cnr}. The average recall is final estimated value used in measuring recall. It is the average of $Recall_1$ and $Recall_2$.

Table C.4 reports the result sets of the queries using adhoc association. The number of resources and relevant resources in these sets are given along with those of the base result sets. The novelty ratio is the proportion of the number of new relevant resources retrieved to the number of relevant resources retrieved. The precision of the adhoc set is the proportion of the number of relevant resources retrieved to the number of resources retrieved.

Table C.1. Proportion of relevant resources found in the review sets

QID	Number of Reviewed Resources in the Result Set	p_d ([L_d:U_d])	Number of Reviewed Resources in the Control Set	p_c ([L_c:U_c])	p_{cnr} ([L_{cnr}:U_{cnr}])
1	77	0.636	500	0.026 [0.015:0.045]	0.012 [0.005:0.027]
2	56	0.464	500	0.002 [0:0.013]	0.002 [0:0.013]
3	84	0.75	500	0.038 [0.024:0.06]	0.036 [0.022:0.057]
4	54	0.556	500	0.032 [0.019:0.053]	0.008 [0.003:0.022]
5	35	0.886	500	0.116 [0.09:0.148]	0.106 [0.081:0.137]
6	5	1	500	0.038 [0.024:0.06]	0.034 [0.021:0.055]
7	348	0.888	500	0.564 [0.519:0.608]	0.008 [0.003:0.022]
8	392	0.737	500	0.42 [0.377:0.465]	0.008 [0.003:0.022]
9	500	0.452 [0.408:0.497]	500	0.084 [0.062:0.113]	0 [0:0.01]
10	18	0.5	500	0.024 [0.013:0.043]	0.02 [0.01:0.038]
11	337	0.828	500	0.458 [0.414:0.503]	0.002 [0:0.013]
12	500	0.172 [0.141:0.209]	500	0.118 [0.092:0.15]	0.004 [0.001:0.016]
13	2	1	500	0.038 [0.024:0.06]	0.036 [0.022:0.057]
14	500	0.026 [0.015:0.045]	500	0.028 [0.016:0.048]	0.008 [0.003:0.022]
15	26	0.038	500	0.002 [0:0.013]	0 [0:0.01]
16	28	0.857	500	0.056 [0.038:0.081]	0.028 [0.016:0.048]
17	18	0.222	500	0.004 [0.001:0.016]	0 [0:0.01]
18	95	0.642	500	0.06 [0.042:0.086]	0.044 [0.028:0.067]
19	6	0.667	500	0.042 [0.027:0.065]	0.036 [0.022:0.057]
20	500	0.852 [0.817:0.881]	500	0.804 [0.766:0.837]	0 [0:0.01]
21	104	0.394	500	0.024 [0.013:0.043]	0 [0:0.01]
22	1	1	137	0.124	0.117
23	67	0.851	500	0.024 [0.013:0.043]	0.014 [0.006:0.03]
24	61	0.77	500	0.05 [0.033:0.074]	0.042 [0.027:0.065]
25	55	0.164	500	0.004 [0.001:0.016]	0.002 [0:0.013]
26	16	0.812	500	0.086 [0.064:0.115]	0.076 [0.055:0.104]
27	500	0.514 [0.469:0.559]	500	0.288 [0.249:0.33]	0.002 [0:0.013]
28	500	0.446 [0.402:0.491]	500	0.33 [0.289:0.373]	0.012 [0.005:0.027]
29	35	0.8	500	0.03 [0.018:0.05]	0.026 [0.015:0.045]
30	148	0.297	500	0.066 [0.047:0.092]	0.052 [0.035:0.076]
31	500	0.568 [0.523:0.612]	500	0.38 [0.338:0.424]	0.008 [0.003:0.022]

QID	Number of Reviewed Resources in the Result Set	p_d ([L_d:U_d])	Number of Reviewed Resources in the Control Set	p_c ([L_c:U_c])	p_{cnr} ([L_{cnr}:U_{cnr}])
32	27	0.926	500	0.006 [0.002:0.019]	0.002 [0:0.013]
33	22	1	77	0.286	0
34	55	1	77	0.714	0
35	5	1	81	0.074	0.012
36	5	1	81	0.062	0
37	3	1	77	0.039	0
38	5	1	77	0.065	0
39	4	1	77	0.052	0
40	500	0.878 [0.845:0.905]	500	0.816 [0.779:0.848]	0.126 [0.099:0.159]
41	85	0.871	500	0.05 [0.033:0.074]	0.002 [0:0.013]
42	500	0.864 [0.83:0.892]	500	0.83 [0.794:0.861]	0.148 [0.119:0.183]
43	500	0.818 [0.781:0.85]	500	0.782 [0.743:0.817]	0.128 [0.101:0.161]
44	69	0.957	500	0.122 [0.095:0.155]	0.066 [0.047:0.092]
45	14	0.929	500	0.012 [0.005:0.027]	0.006 [0.002:0.019]
46	19	1	500	0.022 [0.012:0.04]	0.002 [0:0.013]
47	273	0.934	500	0.198 [0.165:0.236]	0.002 [0:0.013]
48	74	0.986	287	0.798	0.544
49	38	0.947	287	0.474	0.348
50	103	0.99	287	0.927	0.571
51	9	0.667	287	0.23	0.209
52	6	1	287	0.042	0.021
53	15	1	287	0.111	0.059
54	341	0.977	500	0.708 [0.666:0.747]	0.348 [0.307:0.392]
55	29	1	500	0.05 [0.033:0.074]	0.016 [0.007:0.033]
56	63	0.984	500	0.114 [0.088:0.146]	0.042 [0.027:0.065]
57	30	1	500	0.042 [0.027:0.065]	0.008 [0.003:0.022]
58	37	0.919	500	0.068 [0.048:0.095]	0.018 [0.009:0.035]
59	12	0.75	500	0.02 [0.01:0.038]	0.006 [0.002:0.019]
60	10	0.7	500	0.026 [0.015:0.045]	0.018 [0.009:0.035]
61	32	0.938	500	0.066 [0.047:0.092]	0.03 [0.018:0.05]
62	53	0.906	500	0.11 [0.085:0.142]	0.056 [0.038:0.081]
63	127	0.984	500	0.156 [0.126:0.191]	0.012 [0.005:0.027]

Table C.2. Measurements and estimations of the control sets

QID	Total Resources	Actual Relevant or Estimated[1] [Lower:Upper]	Actual Relevant Non-retrieved or Estimated [Lower:Upper]	Control Set Precision or Estimated [Lower:Upper]	Control Set F-measure	Deduction Impact index
1	7,455	194 [109:338]	90 [37:204]	0.026 [0.015:0.045]	0.051	0.359
2	7,566	16 [1:98]	16 [1:98]	0.002 [0:0.013]	0.004	0.586
3	14,691	559 [348:879]	529 [325:844]	0.038 [0.024:0.06]	0.073	0.118
4	1,509	49 [29:80]	13 [4:33]	0.032 [0.019:0.053]	0.062	0.539
5	3,376	392 [304:501]	358 [274:464]	0.116 [0.09:0.148]	0.208	-0.062
6	686	27 [19:42]	24 [17:38]	0.038 [0.024:0.06]	0.073	0.230
7	553	312 [288:337]	5 [4:13]	0.564 [0.519:0.608]	0.721	0.214
8	679	286 [256:316]	6 [4:15]	0.42 [0.377:0.465]	0.592	0.254
9	3,260	274 [202:368]	0 [0:31]	0.084 [0.062:0.113]	0.155	0.468
10	2,677	65 [35:115]	54 [28:101]	0.024 [0.013:0.043]	0.047	0.173
11	624	286 [259:314]	2 [1:9]	0.458 [0.414:0.503]	0.628	0.271
12	843	100 [78:127]	4 [2:14]	0.118 [0.092:0.15]	0.211	0.082
13	817	32 [20:49]	30 [19:47]	0.038 [0.024:0.06]	0.073	0.044
14	817	23 [14:39]	7 [4:18]	0.028 [0.016:0.048]	0.054	-0.004
15	2,181	5 [1:29]	0 [0:21]	0.002 [0:0.013]	0.004	0.068
16	985	56 [38:80]	28 [16:47]	0.056 [0.038:0.081]	0.106	0.480
17	985	4 [2:16]	0 [0:10]	0.004 [0.001:0.016]	0.008	0.356
18	8,056	484 [335:689]	355 [230:539]	0.06 [0.042:0.086]	0.113	0.112
19	1,136	48 [31:74]	41 [26:66]	0.042 [0.027:0.065]	0.081	0.072
20	1,807	1,453 [1,384:1,514]	0 [0:18]	0.804 [0.766:0.837]	0.891	0.028
21	1,807	44 [24:78]	0 [0:18]	0.024 [0.013:0.043]	0.047	0.513
22	137	17	16	0.124	0.221	-0.110
23	3,201	77 [42:137]	45 [20:96]	0.024 [0.013:0.043]	0.047	0.690
24	5,271	264 [176:390]	222 [142:341]	0.05 [0.033:0.074]	0.095	0.192

[1] The estimated number of relevant resources can not be smaller than the actual number of relevant resources found in the result set. Thus, when the number is smaller, the number of relevant resources found in the result set is used instead of this number when measuring recall.

QID	Total Resources	Actual Relevant or Estimated[1] [Lower:Upper]	Actual Relevant Non-retrieved or Estimated [Lower:Upper]	Control Set Precision or Estimated [Lower:Upper]	Control Set F-measure	Deduction Impact index
25	4,310	18 [3:69]	9 [1:56]	0.004 [0.001:0.016]	0.008	0.239
26	2,846	245 [182:328]	217 [157:296]	0.086 [0.064:0.115]	0.158	-0.056
27	2,846	820 [709:940]	6 [1:37]	0.288 [0.249:0.33]	0.447	0.231
28	5,348	1,765 [1,547:1,997]	65 [27:146]	0.33 [0.289:0.373]	0.496	0.105
29	6,701	202 [118:336]	175 [98:303]	0.03 [0.018:0.05]	0.058	0.178
30	7,754	512 [361:717]	404 [271:592]	0.066 [0.047:0.092]	0.124	0.017
31	5,315	2,020 [1,795:2,256]	43 [14:116]	0.38 [0.338:0.424]	0.551	0.165
32	2,512	16 [4:48]	6 [1:33]	0.006 [0.002:0.019]	0.012	0.903
33	77	22	0	0.286	0.444	0.556
34	77	55	0	0.714	0.833	0.167
35	81	6	1	0.074	0.138	0.771
36	81	5	0	0.062	0.116	0.884
37	77	3	0	0.039	0.075	0.925
38	77	5	0	0.065	0.122	0.878
39	77	4	0	0.052	0.099	0.901
40	1,392	1,136 [1,084:1,182]	176 [138:222]	0.816 [0.779:0.848]	0.899	-0.035
41	1,392	70 [47:103]	3 [1:18]	0.05 [0.033:0.074]	0.095	0.827
42	1,392	1,156 [1,105:1,199]	207 [166:255]	0.83 [0.794:0.861]	0.907	-0.064
43	1,392	1,089 [1,034:1,138]	179 [141:225]	0.782 [0.743:0.817]	0.878	-0.053
44	1,365	167 [131:212]	91 [64:127]	0.122 [0.095:0.155]	0.217	0.354
45	1,365	17 [7:38]	9 [3:26]	0.012 [0.005:0.027]	0.024	0.760
46	1,365	31 [16:55]	3 [1:18]	0.022 [0.012:0.04]	0.043	0.806
47	1,365	271 [225:323]	3 [1:18]	0.198 [0.165:0.236]	0.331	0.619
48	287	229	156	0.798	0.888	-0.406
49	287	136	100	0.474	0.643	-0.229
50	287	266	164	0.927	0.962	-0.409
51	287	66	60	0.23	0.374	-0.214
52	287	12	6	0.042	0.080	0.586
53	287	32	17	0.111	0.201	0.438
54	879	623 [586:657]	306 [270:345]	0.708 [0.666:0.747]	0.829	-0.144
55	879	44 [30:65]	15 [8:29]	0.05 [0.033:0.074]	0.095	0.699

QID	Total Resources	Actual Relevant or Estimated[1] [Lower:Upper]	Actual Relevant Non-retrieved or Estimated [Lower:Upper]	Control Set Precision or Estimated [Lower:Upper]	Control Set F-measure	Deduction Impact index
56	879	101 [78:129]	37 [24:57]	0.114 [0.088:0.146]	0.205	0.556
57	879	37 [24:57]	8 [4:20]	0.042 [0.027:0.065]	0.081	0.808
58	879	60 [43:84]	16 [9:31]	0.068 [0.048:0.095]	0.127	0.615
59	879	18 [10:34]	6 [3:17]	0.02 [0.01:0.038]	0.039	0.595
60	879	23 [13:40]	16 [9:31]	0.026 [0.015:0.045]	0.051	0.374
61	879	59 [41:82]	27 [16:45]	0.066 [0.047:0.092]	0.124	0.543
62	879	97 [75:125]	50 [34:72]	0.11 [0.085:0.142]	0.198	0.440
63	879	138 [111:169]	11 [6:24]	0.156 [0.126:0.191]	0.270	0.677

Table C.3. Measurements and estimations of the result sets

QID	Total Retrieved	Total Relevant Resources Retrieved	Precision or Estimated [Lower:Upper]	$Recall_1$ or Estimated [Lower:Upper]	$Recall_2$ or Estimated [Lower:Upper]	Average Recall or Estimated [Lower:Upper]
1	77	49	0.636	0.253 [0.145:0.45]	0.353 [0.194:0.57]	0.303 [0.194:0.45]
2	56	26	0.464	1 [0.265:1]	0.619 [0.21:0.963]	0.81 [0.265:0.963]
3	84	63	0.75	0.113 [0.072:0.181]	0.106 [0.069:0.162]	0.11 [0.072:0.162]
4	54	30	0.556	0.612 [0.375:1]	0.698 [0.476:0.882]	0.655 [0.476:0.882]
5	35	31	0.886	0.079 [0.062:0.102]	0.08 [0.063:0.102]	0.079 [0.063:0.102]
6	5	5	1	0.185 [0.119:0.263]	0.172 [0.116:0.227]	0.179 [0.119:0.227]
7	348	309	0.888	0.99 [0.917:1]	0.984 [0.96:0.987]	0.987 [0.96:0.987]
8	392	289	0.737	1 [0.915:1]	0.98 [0.951:0.986]	0.99 [0.951:0.986]
9	647	293 [264:322]	0.452 [0.408:0.497]	1 [0.717:1]	1 [0.895:1]	1 [0.895:1]
10	18	9	0.5	0.138 [0.078:0.257]	0.143 [0.082:0.243]	0.141 [0.082:0.243]
11	337	279	0.828	0.976 [0.889:1]	0.993 [0.969:0.996]	0.984 [0.969:0.996]
12	599	104 [86:125]	0.172 [0.141:0.209]	1 [0.677:1]	0.963 [0.86:0.984]	0.981 [0.86:0.984]
13	2	2	1	0.062 [0.041:0.1]	0.062 [0.041:0.095]	0.062 [0.041:0.095]
14	774	21 [13:35]	0.026 [0.015:0.045]	0.913 [0.333:1]	0.75 [0.419:0.897]	0.832 [0.419:0.897]
15	26	1	0.038	0.2 [0.034:1]	1 [0.045:1]	0.6 [0.045:1]
16	28	24	0.857	0.429 [0.3:0.632]	0.462 [0.338:0.6]	0.445 [0.338:0.6]
17	18	4	0.222	1 [0.25:1]	1 [0.286:1]	1 [0.286:1]
18	95	61	0.642	0.126 [0.089:0.182]	0.147 [0.102:0.21]	0.136 [0.102:0.182]
19	6	4	0.667	0.083 [0.054:0.129]	0.089 [0.057:0.133]	0.086 [0.057:0.129]
20	1,698	1,447 [1,388:1,497]	0.852 [0.817:0.881]	0.996 [0.917:1]	1 [0.987:1]	0.998 [0.987:1]
21	104	41	0.394	0.932 [0.526:1]	1 [0.695:1]	0.966 [0.695:1]
22	1	1	1	0.059	0.059	0.059
23	67	57	0.851	0.74 [0.416:1]	0.559 [0.373:0.74]	0.65 [0.416:0.74]
24	61	47	0.77	0.178 [0.121:0.267]	0.175 [0.121:0.249]	0.176 [0.121:0.249]
25	55	9	0.164	0.5 [0.13:1]	0.5 [0.138:0.9]	0.5 [0.138:0.9]
26	16	13	0.812	0.053 [0.04:0.071]	0.057 [0.042:0.076]	0.055 [0.042:0.071]
27	1,815	933 [852:1,014]	0.514 [0.469:0.559]	1 [0.906:1]	0.994 [0.958:0.999]	0.997 [0.958:0.999]
28	3,517	1,569 [1,414:1,727]	0.446 [0.402:0.491]	0.889 [0.708:1]	0.96 [0.906:0.985]	0.925 [0.906:0.985]
29	35	28	0.8	0.139 [0.083:0.237]	0.138 [0.085:0.222]	0.138 [0.085:0.222]
30	148	44	0.297	0.086 [0.061:0.122]	0.098 [0.069:0.14]	0.092 [0.069:0.122]
31	3,390	1,926 [1,774:2,074]	0.568 [0.523:0.612]	0.953 [0.786:1]	0.978 [0.939:0.993]	0.966 [0.939:0.993]

QID	Total Retrieved	Total Relevant Resources Retrieved	Precision or Estimated [Lower:Upper]	$Recall_1$ or Estimated [Lower:Upper]	$Recall_2$ or Estimated [Lower:Upper]	Average Recall or Estimated [Lower:Upper]
32	27	25	0.926	1 [0.521:1]	0.806 [0.431:0.962]	0.903 [0.521:0.962]
33	22	22	1	1	1	1
34	55	55	1	1	1	1
35	5	5	1	0.833	0.833	0.833
36	5	5	1	1	1	1
37	3	3	1	1	1	1
38	5	5	1	1	1	1
39	4	4	1	1	1	1
40	1,104	970 [934:999]	0.878 [0.845:0.905]	0.854 [0.79:0.922]	0.846 [0.808:0.879]	0.85 [0.808:0.879]
41	85	74	0.871	1 [0.718:1]	0.961 [0.804:0.987]	0.981 [0.804:0.987]
42	1,104	954 [917:985]	0.864 [0.83:0.892]	0.825 [0.765:0.891]	0.822 [0.782:0.856]	0.823 [0.782:0.856]
43	1,104	904 [862:939]	0.818 [0.781:0.85]	0.83 [0.757:0.908]	0.835 [0.793:0.869]	0.832 [0.793:0.869]
44	69	66	0.957	0.395 [0.311:0.504]	0.42 [0.342:0.508]	0.408 [0.342:0.504]
45	14	13	0.929	0.765 [0.342:1]	0.591 [0.333:0.812]	0.678 [0.342:0.812]
46	19	19	1	0.613 [0.345:1]	0.864 [0.514:0.95]	0.738 [0.514:0.95]
47	273	255	0.934	0.941 [0.789:1]	0.988 [0.934:0.996]	0.965 [0.934:0.996]
48	74	73	0.986	0.319	0.319	0.319
49	38	36	0.947	0.265	0.265	0.265
50	103	102	0.99	0.383	0.383	0.383
51	9	6	0.667	0.091	0.091	0.091
52	6	6	1	0.5	0.5	0.5
53	15	15	1	0.469	0.469	0.469
54	341	333	0.977	0.535 [0.507:0.568]	0.521 [0.491:0.552]	0.528 [0.507:0.552]
55	29	29	1	0.659 [0.446:0.967]	0.659 [0.5:0.784]	0.659 [0.5:0.784]
56	63	62	0.984	0.614 [0.481:0.795]	0.626 [0.521:0.721]	0.62 [0.521:0.721]
57	30	30	1	0.811 [0.526:1]	0.789 [0.6:0.882]	0.8 [0.6:0.882]
58	37	34	0.919	0.567 [0.405:0.791]	0.68 [0.523:0.791]	0.623 [0.523:0.791]
59	12	9	0.75	0.5 [0.265:0.9]	0.6 [0.346:0.75]	0.55 [0.346:0.75]
60	10	7	0.7	0.304 [0.175:0.538]	0.304 [0.184:0.438]	0.304 [0.184:0.438]
61	32	30	0.938	0.508 [0.366:0.732]	0.526 [0.4:0.652]	0.517 [0.4:0.652]
62	53	48	0.906	0.495 [0.384:0.64]	0.49 [0.4:0.585]	0.492 [0.4:0.585]
63	127	125	0.984	0.906 [0.74:1]	0.919 [0.839:0.954]	0.912 [0.839:0.954]

Table C.4. Measurements of the result sets for the queries using adhoc associations

Base QID	Adhoc QID	Total Retrieved	Total Relevant Retrieved	New Relevant Retrieved	Novelty Ratio	Precision	Assigned Association Degree (n)
6	-	5	5	-	-	1.000	-
6	64	7	6	1	0.167	0.857	2
6	65	5	5	0	0.000	1.000	3
48	-	74	73	-	-	0.986	-
48	66	93	89	16	0.18	0.957	2
48	67	81	79	6	0.076	0.975	3
49	-	38	36	-	-	0.947	-
49	68	46	41	5	0.122	0.891	2
49	69	42	39	3	0.077	0.929	3
50	-	103	102	-	-	0.990	-
50	70	129	120	18	0.15	0.930	2
50	71	116	113	11	0.097	0.974	3
54	-	341	333	-	-	0.977	-
54	72	466	422	89	0.211	0.906	2
54	73	394	371	38	0.102	0.942	3
55	-	29	29	-	-	1.000	-
55	74	35	33	4	0.121	0.943	2
55	75	30	30	1	0.033	1.000	3

APPENDIX D

CHI-SQUARE TESTS OF INDEPENDENCE

This section reports the results from the chi-square test of independence on the defined independent variables and the precision and recall of the result sets. Each variable is grouped as shown in section D.1. The test results are reported in section D.2 and D.3. The null hypothesis on the independence between the variables uses the significant level of 0.05 (p-value < 0.05).

D.1 VARIABLES

Precision	
0	**1**
0-0.8	>0.8

Recall	
0	**1**
0-0.8	>0.8

Total Resources			
0	**1**	**2**	**3**
0-500	501-1,000	1001-5,000	>5,000

Total Relevant			
0	**1**	**2**	**3**
0-25	25-100	101-1,000	>1,000

Total Subcategories		
0	**1**	**2**
0	1-10	>10

Average Subcategories		
0	**1**	**2**
0	1-10	>10

Average Subcategory Max Depth		
0	**1**	**2**
0	1	>1

Average Members		
0	**1**	**2**
0-100	101-1,000	>1,000

Total Connectives			
0	**1**	**2**	**3**
1	2	3	>3

Conjunction Terms	
0	**1**
1	>1

Disjunction Terms	
0	**1**
None	1+

Negation Terms	
0	**1**
None	1+

Quantifier/ Cardinality Terms		
0	**1**	**2**
None	1	>1

Judges' Relevant Agreement		
0	**1**	**2**
0-0.5	0.51-0.8	0.81-1

D.2 CHI-SQUARE TESTS OF INDEPENDENCE ON PRECISION

Total Resources * Precision Crosstabulation

Count

		Precision		
		0	1	Total
Total Resources	0	1	13	14
	1	6	13	19
	2	8	13	21
	3	9		9
Total		24	39	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	20.655[a]	3	.000
Likelihood Ratio	24.917	3	.000
Linear-by-Linear Association	16.364	1	.000
N of Valid Cases	63		

a. 1 cells (12.5%) have expected count less than 5. The minimum expected count is 3.43.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	-.458	.088	-4.867	.000
N of Valid Cases		63			

Total Relevant * Precision Crosstabulation

Count

		Precision		
		0	1	Total
Total Relevant	0	5	9	14
	1	7	15	22
	2	8	10	18
	3	4	5	9
Total		24	39	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	.863[a]	3	.834
Likelihood Ratio	.864	3	.834
Linear-by-Linear Association	.475	1	.491
N of Valid Cases	63		

a. 1 cells (12.5%) have expected count less than 5. The minimum expected count is 3.43.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	-.083	.116	-.710	.478
N of Valid Cases		63			

Total Subcat. * Precision Crosstabulation

Count

		Precision		
		0	1	Total
Total Subcat.	0	9	30	39
	1	8	3	11
	2	7	6	13
Total		24	39	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	10.692[a]	2	.005
Likelihood Ratio	10.759	2	.005
Linear-by-Linear Association	6.318	1	.012
N of Valid Cases	63		

a. 2 cells (33.3%) have expected count less than 5. The minimum expected count is 4.19.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	-.340	.116	-2.887	.004
N of Valid Cases		63			

Average Subcat. * Precision Crosstabulation

Count

		Precision		
		0	1	Total
Average Subcat.	0	9	30	39
	1	10	5	15
	2	5	4	9
Total		24	39	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	10.086[a]	2	.006
Likelihood Ratio	10.134	2	.006
Linear-by-Linear Association	6.831	1	.009
N of Valid Cases	63		

a. 1 cells (16.7%) have expected count less than 5. The minimum expected count is 3.43.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	-.354	.116	-2.999	.003
N of Valid Cases		63			

Avg Subcat Depth * Precision Crosstabulation

Count

		Precision		
		0	1	Total
Avg Subcat Depth	0	9	30	39
	1	12	9	21
	2	3		3
Total		24	39	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	11.836[a]	2	.003
Likelihood Ratio	12.913	2	.002
Linear-by-Linear Association	11.589	1	.001
N of Valid Cases	63		

a. 2 cells (33.3%) have expected count less than 5. The minimum expected count is 1.14.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	-.410	.112	-3.446	.001
N of Valid Cases		63			

Average Members * Precision Crosstabulation

Count

		Precision		
		0	1	Total
Average Members	0		8	8
	1	13	21	34
	2	11	10	21
Total		24	39	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	6.741[a]	2	.034
Likelihood Ratio	9.432	2	.009
Linear-by-Linear Association	5.799	1	.016
N of Valid Cases	63		

a. 2 cells (33.3%) have expected count less than 5. The minimum expected count is 3.05.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	-.282	.104	-2.601	.009
N of Valid Cases		63			

Total Connectives * Precision Crosstabulation

Count

		Precision		
		0	1	Total
Total Connectives	0	5	4	9
	1	7	12	19
	2	2	12	14
	3	10	11	21
Total		24	39	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	5.349[a]	3	.148
Likelihood Ratio	5.809	3	.121
Linear-by-Linear Association	.047	1	.827
N of Valid Cases	63		

a. 1 cells (12.5%) have expected count less than 5. The minimum expected count is 3.43.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	.015	.124	.125	.901
N of Valid Cases		63			

Conjunction * Precision Crosstabulation

Count

		Precision		
		0	1	Total
Conjunction	0	15	29	44
	1	9	10	19
Total		24	39	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)	Exact Sig. (2-sided)	Exact Sig. (1-sided)
Pearson Chi-Square	.992[b]	1	.319		
Continuity Correction[a]	.509	1	.476		
Likelihood Ratio	.980	1	.322		
Fisher's Exact Test				.400	.237
Linear-by-Linear Association	.976	1	.323		
N of Valid Cases	63				

a. Computed only for a 2x2 table

b. 0 cells (.0%) have expected count less than 5. The minimum expected count is 7.24.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	-.125	.128	-.978	.328
N of Valid Cases		63			

Disjunction * Precision Crosstabulation

Count

		Precision		
		0	1	Total
Disjunction	0	16	28	44
	1	8	11	19
Total		24	39	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)	Exact Sig. (2-sided)	Exact Sig. (1-sided)
Pearson Chi-Square	.185[b]	1	.667		
Continuity Correction[a]	.022	1	.882		
Likelihood Ratio	.184	1	.668		
Fisher's Exact Test				.779	.438
Linear-by-Linear Association	.183	1	.669		
N of Valid Cases	63				

a. Computed only for a 2x2 table

b. 0 cells (.0%) have expected count less than 5. The minimum expected count is 7.24.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	-.054	.127	-.426	.670
N of Valid Cases		63			

Negation * Precision Crosstabulation

Count

		Precision		
		0	1	Total
Negation	0	14	34	48
	1	10	5	15
Total		24	39	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)	Exact Sig. (2-sided)	Exact Sig. (1-sided)
Pearson Chi-Square	6.815[b]	1	.009		
Continuity Correction[a]	5.318	1	.021		
Likelihood Ratio	6.686	1	.010		
Fisher's Exact Test				.014	.011
Linear-by-Linear Association	6.707	1	.010		
N of Valid Cases	63				

a. Computed only for a 2x2 table

b. 0 cells (.0%) have expected count less than 5. The minimum expected count is 5.71.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	-.329	.123	-2.500	.012
N of Valid Cases		63			

Quant./Card. * Precision Crosstabulation

Count

		Precision		
		0	1	Total
Quant./Card.	0	21	11	32
	1		16	16
	2	3	12	15
Total		24	39	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	22.213[a]	2	.000
Likelihood Ratio	27.535	2	.000
Linear-by-Linear Association	13.066	1	.000
N of Valid Cases	63		

a. 0 cells (.0%) have expected count less than 5. The minimum expected count is 5.71.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	.472	.104	4.463	.000
N of Valid Cases		63			

Judge Agreement * Precision Crosstabulation

Count

		Precision		
		0	1	Total
Judge Agreement	0	18	7	25
	1	6	16	22
	2		16	16
Total		24	39	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	23.125[a]	2	.000
Likelihood Ratio	28.301	2	.000
Linear-by-Linear Association	22.311	1	.000
N of Valid Cases	63		

a. 0 cells (.0%) have expected count less than 5. The minimum expected count is 6.10.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	.571	.076	6.886	.000
N of Valid Cases		63			

D.3 CHI-SQUARE TESTS OF INDEPENDENCE ON RECALL

Total Resources * Recall Crosstabulation

Count

		Recall		
		0	1	Total
Total Resources	0	7	7	14
	1	11	8	19
	2	11	10	21
	3	6	3	9
Total		35	28	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	.753[a]	3	.861
Likelihood Ratio	.763	3	.858
Linear-by-Linear Association	.291	1	.590
N of Valid Cases	63		

a. 1 cells (12.5%) have expected count less than 5. The minimum expected count is 4.00.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	-.058	.115	-.504	.614
N of Valid Cases		63			

Total Relevant * Recall Crosstabulation

Count

		Recall		
		0	1	Total
Total Relevant	0	7	7	14
	1	15	7	22
	2	11	7	18
	3	2	7	9
Total		35	28	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	5.870[a]	3	.118
Likelihood Ratio	6.036	3	.110
Linear-by-Linear Association	1.178	1	.278
N of Valid Cases	63		

a. 1 cells (12.5%) have expected count less than 5. The minimum expected count is 4.00.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	.112	.120	.934	.350
N of Valid Cases		63			

Total Subcat. * Recall Crosstabulation

Count

		Recall		
		0	1	Total
Total Subcat.	0	22	17	39
	1	4	7	11
	2	9	4	13
Total		35	28	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	2.637[a]	2	.268
Likelihood Ratio	2.666	2	.264
Linear-by-Linear Association	.201	1	.654
N of Valid Cases	63		

a. 1 cells (16.7%) have expected count less than 5. The minimum expected count is 4.89.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	-.025	.120	-.211	.833
N of Valid Cases		63			

Average Subcat. * Recall Crosstabulation

Count

		Recall		
		0	1	Total
Average Subcat.	0	22	17	39
	1	6	9	15
	2	7	2	9
Total		35	28	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	3.282[a]	2	.194
Likelihood Ratio	3.409	2	.182
Linear-by-Linear Association	.328	1	.567
N of Valid Cases	63		

a. 1 cells (16.7%) have expected count less than 5. The minimum expected count is 4.00.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	-.029	.120	-.244	.807
N of Valid Cases		63			

Avg Subcat Depth * Recall Crosstabulation

Count

		Recall		
		0	1	Total
Avg Subcat Depth	0	22	17	39
	1	11	10	21
	2	2	1	3
Total		35	28	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	.247[a]	2	.884
Likelihood Ratio	.251	2	.882
Linear-by-Linear Association	.000	1	1.000
N of Valid Cases	63		

a. 2 cells (33.3%) have expected count less than 5. The minimum expected count is 1.33.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	.012	.123	.098	.922
N of Valid Cases		63			

Average Members * Recall Crosstabulation

Count

		Recall		
		0	1	Total
Average Members	0	1	7	8
	1	23	11	34
	2	11	10	21
Total		35	28	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	8.105[a]	2	.017
Likelihood Ratio	8.658	2	.013
Linear-by-Linear Association	1.169	1	.280
N of Valid Cases	63		

a. 2 cells (33.3%) have expected count less than 5. The minimum expected count is 3.56.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	-.102	.127	-.796	.426
N of Valid Cases		63			

Total Connectives * Recall Crosstabulation

Count

		Recall		
		0	1	Total
Total Connectives	0	9		9
	1	9	10	19
	2	5	9	14
	3	12	9	21
Total		35	28	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	9.969[a]	3	.019
Likelihood Ratio	13.339	3	.004
Linear-by-Linear Association	2.069	1	.150
N of Valid Cases	63		

a. 1 cells (12.5%) have expected count less than 5. The minimum expected count is 4.00.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	.152	.112	1.351	.177
N of Valid Cases		63			

Conjunction * Recall Crosstabulation

Count

		Recall		
		0	1	Total
Conjunction	0	26	18	44
	1	9	10	19
Total		35	28	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)	Exact Sig. (2-sided)	Exact Sig. (1-sided)
Pearson Chi-Square	.739[b]	1	.390		
Continuity Correction[a]	.340	1	.560		
Likelihood Ratio	.736	1	.391		
Fisher's Exact Test				.421	.279
Linear-by-Linear Association	.727	1	.394		
N of Valid Cases	63				

a. Computed only for a 2x2 table

b. 0 cells (.0%) have expected count less than 5. The minimum expected count is 8.44.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	.108	.126	.855	.392
N of Valid Cases		63			

Disjunction * Recall Crosstabulation

Count

		Recall		
		0	1	Total
Disjunction	0	25	19	44
	1	10	9	19
Total		35	28	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)	Exact Sig. (2-sided)	Exact Sig. (1-sided)
Pearson Chi-Square	.094[b]	1	.759		
Continuity Correction[a]	.001	1	.976		
Likelihood Ratio	.094	1	.759		
Fisher's Exact Test				.788	.486
Linear-by-Linear Association	.093	1	.761		
N of Valid Cases	63				

a. Computed only for a 2x2 table

b. 0 cells (.0%) have expected count less than 5. The minimum expected count is 8.44.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	.039	.126	.306	.760
N of Valid Cases		63			

Negation * Recall Crosstabulation

Count

		Recall		
		0	1	Total
Negation	0	35	13	48
	1		15	15
Total		35	28	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)	Exact Sig. (2-sided)	Exact Sig. (1-sided)
Pearson Chi-Square	24.609[b]	1	.000		
Continuity Correction[a]	21.745	1	.000		
Likelihood Ratio	30.485	1	.000		
Fisher's Exact Test				.000	.000
Linear-by-Linear Association	24.219	1	.000		
N of Valid Cases	63				

a. Computed only for a 2x2 table

b. 0 cells (.0%) have expected count less than 5. The minimum expected count is 6.67.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	.625	.075	5.612	.000
N of Valid Cases		63			

Quant./Card. * Recall Crosstabulation

Count

		Recall		
		0	1	Total
Quant./Card.	0	18	14	32
	1	6	10	16
	2	11	4	15
Total		35	28	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	4.039[a]	2	.133
Likelihood Ratio	4.130	2	.127
Linear-by-Linear Association	.562	1	.454
N of Valid Cases	63		

a. 0 cells (.0%) have expected count less than 5. The minimum expected count is 6.67.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	-.066	.118	-.553	.581
N of Valid Cases		63			

Judge Agreement * Recall Crosstabulation

Count

		Recall		
		0	1	Total
Judge Agreement	0	15	10	25
	1	12	10	22
	2	8	8	16
Total		35	28	63

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	.409[a]	2	.815
Likelihood Ratio	.409	2	.815
Linear-by-Linear Association	.401	1	.526
N of Valid Cases	63		

a. 0 cells (.0%) have expected count less than 5. The minimum expected count is 7.11.

		Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Ordinal by Ordinal	Kendall's tau-b	.076	.119	.642	.521
N of Valid Cases		63			

APPENDIX E

RELEVANCE JUDGMENT

E.1 JUDGES

The three recruited judges were the graduates of the Master of Library and Information Science program (MLIS) in the School of Information Sciences at the University of Pittsburgh. The judges were recommended as knowledgeable in subject classification and cataloging by Professor Arlene Taylor, a professor emeritus of the School of Information Sciences, who has been teaching in the subjects for decades. Among the three recruited judges, the judge with the most professional experience in cataloging was assigned as the third judge. The judges were employed part-time and were paid in an hourly basis.

E.2 RELEVANCE JUDGMENT TASKS AND TOOLS

In order to facilitate the judges in performing the judgment task, a system was created for the judges. The system provided the Web interfaces for accessing the information of the review resources and allowing the judges in making relevance judgment on the resources. The use of the system in performing the relevance judgment task could be described as follows.

In order to begin a task session, the judge must log on to the system. Once the judge is logged on, the query descriptions are displayed to allow the judge to begin reviewing the

resources for the queries. When the judge selects a query, a resource in the review set will be displayed along with some specific information, such as book title, author names, media format, publish date, publisher (Figure E.1). In addition, the judge can choose to view the book detail information such as table of contents, sample pages and editorial reviews, when they are available from the Amazon.com web site. The judge can use the information to help in making the relevance judgment. Once the judge assesses the relevancy of the resource by selecting from the given choices, the next resource will be displayed. The process is repeated until all resources for the query are reviewed and the judge will be led to the main query selection page.

The judge is given three choices in assessing the relevancy of a resource in the context of a query: *Relevant*, *Not Relevant* and *Not Sure*. The judge was advised to choose *Relevant* for the resource that contains a high level of information related to the stipulated query and to choose *Not Relevant* otherwise. The judge was advised to choose *Unsure* only when the decision could not be made or the judge wants to delay the decision until the end. The resources marked *Unsure* by the judge will appear again for reassessment after all the resources for the query were reviewed. After the reassessment, the resources, which were still marked *Unsure* will be left as is and the judgment task for the query is completed.

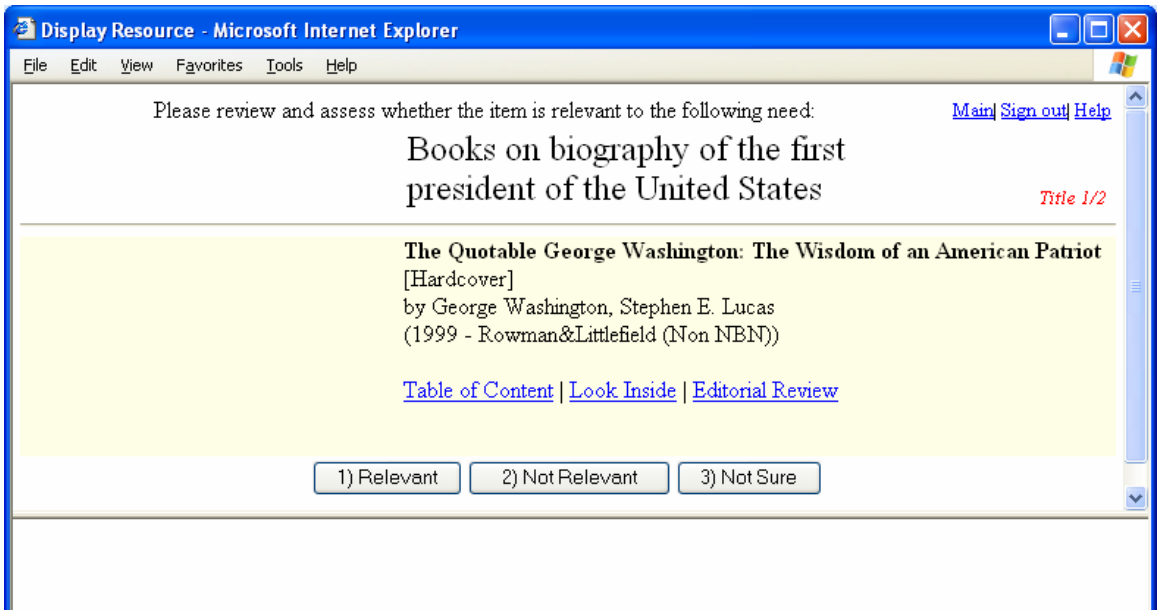


Figure E.1. Relevant Judgment Tool

The judge could perform the task over multiple sessions by resuming the work from previous session. The relevance judgment tasks were conducted over two-month period. The three judges use the same judgment tool and resource information in performing the judgment tasks.

E.3 REPORT ON JUDGE AGREEMENT IN THE RELEVANCE JUDGMENT

The relevance agreement ratio (*Ra*) is the proportion of the total number of resources which both judges agreed on the relevancy to the total number of resources which either judge assessed as relevant. The Disagreed Relevance ratio is the proportion of the total disagreed resources and unsure resources which were assessed as relevant by the third judge. The *Ra* measured for each review set is reported in Table E.1.

Table E.1. Relevance agreement ratio of the review sets

QID	Total Unique Reviewed	Total Relevant Agreed	Total Relevant Disagreed	Total Unsure	Relevance Agreement Ratio (Ra)	Total Disagreed Relevant	Disagreed Relevance Ratio
1	563	39	27	0	0.591	16	0.593
2	551	8	28	0	0.222	19	0.679
3	582	60	75	33	0.444	21	0.194
4	536	23	41	0	0.359	11	0.268
5	529	30	64	1	0.319	54	0.831
6	503	14	15	2	0.483	8	0.471
7	536	192	139	0	0.580	121	0.871
8	608	181	136	0	0.571	112	0.824
9	922	160	82	4	0.661	71	0.826
10	512	13	10	1	0.565	6	0.545
11	562	194	99	1	0.662	86	0.860
12	685	71	173	0	0.291	28	0.162
13	499	15	8	0	0.652	5	0.625

QID	Total Unique Reviewed	Total Relevant Agreed	Total Relevant Disagreed	Total Unsure	Relevance Agreement Ratio (Ra)	Total Disagreed Relevant	Disagreed Relevance Ratio
14	689	9	72	0	0.111	11	0.153
15	519	1	26	0	0.037	0	0.000
16	512	30	11	0	0.732	8	0.727
17	512	2	3	0	0.400	2	0.667
18	583	10	73	0	0.120	73	1.000
19	501	5	17	0	0.227	17	1.000
20	861	423	390	3	0.520	286	0.728
21	575	28	31	0	0.475	13	0.419
22	137	16	11	1	0.593	1	0.083
23	562	21	45	0	0.318	43	0.956
24	556	28	54	0	0.341	40	0.741
25	547	4	15	0	0.211	6	0.400
26	511	36	23	0	0.610	15	0.652
27	912	72	476	0	0.131	282	0.592
28	951	21	469	27	0.043	343	0.692
29	533	35	48	0	0.422	6	0.125
30	633	14	159	6	0.081	56	0.339
31	945	29	569	10	0.048	414	0.715
32	525	12	20	0	0.375	14	0.700
33	77	22	1	0	0.957	0	0.000
34	77	53	2	0	0.964	2	1.000
35	81	6	0	0	1.000	0	N/A
36	81	5	1	0	0.833	0	0.000
37	77	3	0	1	1.000	0	0.000
38	77	5	0	1	1.000	0	0.000
39	77	4	0	1	1.000	0	0.000
40	812	459	323	0	0.587	225	0.697
41	558	65	18	0	0.783	10	0.556
42	812	407	349	0	0.538	274	0.785
43	805	414	338	0	0.551	233	0.689

QID	Total Unique Reviewed	Total Relevant Agreed	Total Relevant Disagreed	Total Unsure	Relevance Agreement Ratio (Ra)	Total Disagreed Relevant	Disagreed Relevance Ratio
44	538	28	72	0	0.280	71	0.986
45	509	15	2	0	0.882	1	0.500
46	509	18	2	0	0.900	2	1.000
47	666	99	161	0	0.381	157	0.975
48	287	216	16	0	0.931	13	0.813
49	287	61	81	0	0.430	75	0.926
50	287	258	17	0	0.938	8	0.471
51	287	24	87	0	0.216	42	0.483
52	287	11	1	0	0.917	1	1.000
53	287	25	7	0	0.781	7	1.000
54	653	457	58	0	0.887	50	0.862
55	512	35	2	0	0.946	2	1.000
56	526	65	23	0	0.739	18	0.783
57	513	26	8	0	0.765	8	1.000
58	509	37	9	0	0.804	6	0.667
59	505	9	3	0	0.750	3	1.000
60	504	12	9	0	0.571	4	0.444
61	513	35	11	0	0.761	10	0.909
62	522	64	18	0	0.780	12	0.667
63	553	115	16	0	0.878	16	1.000

BIBLIOGRAPHY

[1] Bush, V., "As we may think," *The Atlantic Monthly*, vol. 176, no. 1, pp. 101-108, 1945.

[2] Wool, G., "A meditation on metadata," in Wayne Jones (ed.) *E-Serials: Publishers, Libraries, Users, and Standards* The Haworth Press, Inc., 1998, pp. 167-178.

[3] Raggett, D., Le Hors, A., and Jacobs, I. The global structure of an HTML document. HTML 4.01 Specification W3C Recommendation 24 December 1999 . 1999. Available: http://www.w3.org/TR/REC-html40/struct/global.html

[4] W3C. Platform for Internet Content Selection (PICS). W3C . 2001. Available: http://www.w3.org/PICS/

[5] Dublin Core Metadata Initiative. Dublin Core Metadata Element Set, Version 1.1: Reference Description. Dublin Core Metadata Initiative Recommendation . 1999. Available: http://purl.org/dc/documents/rec-dces-19990702.htm

[6] Lagoze, C., Lynch, C., and Daniel, R. Jr. The Warwick Framework: A container architecture for aggregating sets of metadata. Cornell Computer Science Technical Report TR96-1593 . 1996. Available: http://cs-tr.cs.cornell.edu/Dienst/UI/2.0/Describe/ncstrl.cornell/TR96-1593

[7] Lassila , O. and Swick , R. R. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation 22 February 1999 . 1999. Available: http://www.w3.org/TR/REC-rdf-syntax/

[8] Berners-Lee, T., Hendler, J., and Lassila , O., "The Semantic web," *Scientific American*, pp. 28-37, 2001.

[9] Williamson, N. J. Knowledge structures and the Internet. 23-27. 1997. The Hague, Netherlands. Knowledge Organization for Information Retrieval.

[10] W3C. Semantic Web activity statement. W3C Technology & Society Domain Activity Statement . 2002. Available: http://www.w3.org/2001/sw/Activity

[11] Hamming, R. W., *Coding and Information Theory*, 2 ed. New Jersey: Prentice Hall, 1986.

[12] Debons, A., Horne, E., and Cronenweth, S., *Information Science: An Integrated View* Boston: G.L. Hall & Co., 1988.

[13] Dertouzos, M. L., *What will be : how the new world of information will change our lives*, 1 ed. San Francisco, California: HarperEdge, 1997.

[14] Choo, C. W., Detlor, B., and Turnbull, D., *Web Work: Information seeking and knowledge work on the World Wide Web* Dordrecht/ Boston/ London: Kluwer Academic Publishers, 2000.

[15] Chan, L. M., *Cataloging and Classification: An Introduction* McGraw-Hill, 1981.

[16] The American Heritage® Stedman's Medical Dictionary, *The American Heritage® Stedman's Medical Dictionary*, 4 ed. Houghton Mifflin Company, 2002.

[17] Merriam-Webster Collegiate® Dictionary. Merriam-Webster Collegiate® Dictionary. Merriam-Webster Collegiate® Dictionary . 2003. Available: http://www.m-w.com/

[18] McCarthy, J. Programs with common sense. 1960. H. M. Stationery Office. Proceedings of the Teddington Conference on the Mechanization of Thought Processes. 11-24-1958.

[19] Canfora, L., *The Vanished Library* Berkeley, CA: University of California Press, 1989.

[20] OCLC. Dewey Decimal Classification. Dewey Decimal Classification Home Page (OCLC Forest Press) . 2002. Available: http://www.oclc.org/dewey/

[21] Library of Congress. Library of Congress Classification Outline. Library of Congress . 2002. Available: http://www.loc.gov/catdir/cpso/lcco/lcco.html

[22] Swanson, D. R., "Undiscovered public knowledge," *The Library Quarterly*, vol. 56, no. 2, pp. 103-118, 1986.

[23] Buchanan, M., *Nexus: Small Worlds and the Groundbreaking Science of Networks*, 1 ed. W.W. Norton & Company, 2002.

[24] Berners-Lee, T., *Weaving the Web: The original design and ultimate destiny of the World Wide Web* New York, NY: HarperCollins, 2000.

[25] Berners-Lee, T. Semantic Web. Presentation at XML 2000, December 6, 2000 . 2000. Washington, DC. Available: http://www.w3.org/2000/Talks/1206-xml2k-tbl/

[26] Bray, T., Paoli, J., Sperberg-McQueen, C. M., and Maler, E. Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation 6 October 2000 . 2000. Available: http://www.w3.org/TR/2000/REC-xml-20001006

[27] Decker, S., Melnik, S., Van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., Erdmann, M., and Horrocks, I. The Semantic Web: The roles of XML and RDF. IEEE Internet Computing 4[5], 63-73. 2000.

[28] Brickley, D. and Guha, R. V. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 February 2004 . 2004. Available: http://www.w3.org/TR/rdf-schema

[29] Berners-Lee, T. Rules and Facts: Inference engines vs Web. Personal Note . 2000. Available: http://www.w3.org/DesignIssues/Rules.html

[30] Institute of Electrical and Electronics Engineers, *IEEE standard computer dictionary: A compilation of IEEE standard computer glossaries* New York, NY: 1990.

[31] Brodie, M. L. The promise of distributed computing and the challenge of legacy information systems. 1-31. 1992. Lorne, Australia. Proceedings of the IFIP WG2.6 Database Semantics Conference on Interoperable Database Systems.

[32] Vckovski, A., "Interoperability and spatial information theory," Norwell, Massachusetts: Kluwer Academic Publishers, 1999.

[33] Spring, M. B., "Reference model for data interchange standards," *IEEE Computer*, vol. 29, no. 8, pp. 87-88, 1996.

[34] Melnik, S. and Decker, S. A layered approach to information modeling and interoperability on the Web. 2000. Lisbon, Portugal. Proceedings of the ECDL'00 Workshop on the Semantic Web.

[35] Wiederhold, G., "Mediators in the architecture of future information systems," *IEEE Computer*, vol. 25, no. 3, 1992.

[36] Kashyap, V. and Sheth, A. Semantics-based information brokering. 363-370. 1994. Proceedings of the 3rd International Conference on Information and Knowledge Systems.

[37] Schwarz, P. M. and Roth, M. T. Don't scrap it, wrap it! A wrapper architecture for legacy data sources. 266-275. 1997. Athens, Greece. Proceedings of the 23rd VLDB Conference.

[38] Melnik, S. e. al. Generic Interoperability Framework. Working Paper . 1999. Available: http://www-diglib.stanford.edu/diglib/ginf/WD/ginf-overview/

[39] Davis, R., Schrobe, H., and Szolovits, P., "What is a Knowledge Representation?," *AI Magazine*, vol. 14, no. 1, pp. 17-33, 1993.

[40] Frege, G., "On sense and reference," in Geach, P. and Black, M. (eds.) *Translations from the Philosophical Writings of Gottlob Frege* Oxford: Blackwell, 1892.

[41] Sowa, J. F., *Knowledge Representation: Logical, philosophical, and computational foundations* Pacific Grove, CA: Brooks/Cole, 2000.

[42] Russell, S. and Norvig, P., *Artificial Intelligence: A modern approach* New Jersey: Prentice-Hall, Inc., 1995.

[43] Genesereth, M. R. and Nilsson, N. J., *Logical foundations of artificial intelligence* Los Altos, CA: Morgan Kaufmann Publishers, Inc., 1987.

[44] Robinson, J. A., "A machine-oriented logic based on the resolution principle," *Journal of the Association for Computing Machinery*, vol. 12, no. 1, pp. 23-41, 1965.

[45] Levesque, H. J. and Brachman, R. J., "A fundamental tradeoff in knowledge representation and reasoning (revised version)," in Brachman, R. J. and Levesque, H. J. (eds.) *Readings in Knowledge Representation* San Mateo, CA: Morgan Kaufmann, 1985, pp. 41-70.

[46] Brachman, R. J. and Levesque, H. J. The tractability of subsumption in frame-Based description languages. 1984. Austin, Texas. Proceedings of AAAI-84. 1984.

[47] Gruber, T. R., "A translation approach to portable ontologies," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199-220, 1993.

[48] Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., and Swartout, W. R., "Enabling technology for knowledge sharing," *AI Magazine*, vol. 12, no. 3, pp. 36-56, 1991.

[49] Swartout, B., Patil, R., Knight, K., and Russ, T. Toward distributed use of large-scale ontologies. 1996. Proceedings of the Tenth Knowledge Acquisition for Knowledge - based Systems Workshop.

[50] Minsky, M., "A framework for representing knowledge," in Winston, P. (ed.) *The Psychology of Computer Vision* New York: McGraw-Hill, 1975, pp. 211-277.

[51] Quillian, M. R., "Word concepts: a theory and simulation of some basic semantic capabilities," *Behavioral Science*, no. 12, pp. 410-430, 1967.

[52] Collins, A. M. and Quillian, M. R., "Facilitating retrieval from semantic memory: The effect of repeating part of an inference," in Sanders, A. F. (ed.) *Acta Psychologica 33 Attention and Performance III* Amsterdam: North-Holland Publ., 1970, pp. 304-314.

[53] Woods, W. A., "What's in a link: foundations of semantic networks," in Bobrow, D. G. and Collins, A. M. (eds.) *Representation and Understanding: Studies in Cognitive Science* New York: Academic Press, 1975, pp. 35-82.

[54] Brachman, R. J., "On the epistemological status of semantic networks," in Findler, N. V. (ed.) *Associative Networks: Representation and Use of Knowledge by Computers* New York: Academic Press, 1979, pp. 3-50.

[55] Brachman, R. J. and Schmolze, J. G., "An overview of the KL-ONE knowledge representation system," *Cognitive Science*, vol. 9 pp. 171-216, 1985.

[56] Woods, W. A. and Schmolze, J. G., "The KL-ONE family," *Computers & Mathematics with Applications*, vol. 23, no. 2-5, pp. 133-177, 1992.

[57] Brachman, R. J., Fikes, R. E., and Levesque, H. J., "KRYPTON - A functional-approach to knowledge representation," *Computer*, vol. 16, no. 10, pp. 67-73, 1983.

[58] Schmidt-Schauß, M. and Smolka, G., "Attributive concept descriptions with complements," *Artifitial Intelligence Journal*, vol. 48, no. 1, pp. 1-26, 1991.

[59] Berners-Lee, T. Web architecture from 50,000 feet. Personal Note . 1998. W3C. Available: http://www.w3.org/DesignIssues/Architecture.html

[60] Berners-Lee, T., Fielding, R., and Frystyk, H. Hypertext Transfer Protocol -- HTTP/1.0. RFC 1945 . 1996. IETF Network Working Group. Available: http://www.ietf.org/rfc/rfc1945.txt?number=1945

[61] SGML Users' Group. A brief history of the development of SGML. SGML Users' Group . 1990. Available: http://www.sgmlsource.com/history/sgmlhist.htm

[62] International Organization for Standardization. ISO 8879: Information processing---Text and office systems---Standard Generalized Markup Language (SGML). ISO Standard . 1986. Geneva. Available: http://www.iso.ch/

[63] Sperberg-McQueen, C. M. and Burnard, L. A gentle introduction to SGML. TEI Guidelines for Electronic Text Encoding and Interchange . 2001. Text Encoding Initiative. Available: http://www-tei.uic.edu/orgs/tei/sgml/teip3sg/index.html

[64] Berners-Lee, T. and Connolly, D. Hypertext Markup Language (HTML): A representation of textual information and metaInformation for retrieval and interchange. W3C . 1993. Available: http://www.w3.org/MarkUp/draft-ietf-iiir-html-01.txt

[65] Berners-Lee, T. and Connolly, D. HyperText Markup Language Specification -- 2.0. Internet . 1995. Available: http://www.ics.uci.edu/pub/ietf/html/rfc1866.txt

[66] Raggett, D. HTML 3.2 Reference Specification. 1997. W3C. Available: http://www.w3.org/TR/REC-html32

[67] Raggett, D., Hors, A. L., and Jacobs, I. HTML 4.0 Specification. 1998. W3C. Available: http://www.w3.org/TR/1998/REC-html40-19980424/

[68] Berners-Lee, T., Fielding, R., and Masinter, L. Uniform Resource Identifiers (URI): Generic Syntax. RFC 2396 . 1998. Available: http://www.ietf.org/rfc/rfc2396.txt?number=2396

[69] Bosak, J. XML, Java, and the future of the Web. Online article . 1997. Available: http://www.ibiblio.org/bosak/xml/why/xmlapps.htm

[70] Abiteboul, S., Buneman, P., and Suciu, D., *Data on the Web: From relations to semistructured data and XML* San Francisco, CA: Morgan Kaufmann, 2000.

[71] Clark, J. Comparison of SGML and XML. World Wide Web Consortium Note 15-December-1997 . 1997. Available: http://www.w3.org/TR/NOTE-sgml-xml

[72] W3C DOM Working Group. Document Object Model (DOM). W3C Architecture Domain . 2001. Available: http://www.w3.org/DOM/

[73] Megginson, D. SAX 2.0: The Simple API for XML. Megginson Technologies . 2000. Available: http://www.megginson.com/SAX/index.html

[74] Berners-Lee, T., Connolly, D., and Swick, R. R. Web architecture: Describing and exchanging data. W3C . 6-7-1999. Available: http://www.w3.org/1999/06/07-WebData.html

[75] Thompson, H. S., Maloney, M., and Mendelsohn, N. XML Schema Part 1: Structures. W3C Note . 2000. Available: http://www.w3.org/TR/xmlschema-1/

[76] Biron, P. V. and Malhotra, A. XML Schema Part 2: Datatypes. W3C Note . 2000. Available: http://www.w3.org/TR/xmlschema-2/

[77] Bray, T., Hollander, D., and Layman, A. Namespaces in XML. W3C Recommendation 14 January 1999 . 1999. Available: http://www.w3.org/TR/1999/REC-xml-names-19990114/

[78] DeRose, S., Maler, E., and Orchard, D. XML Linking Language (XLink) Version 1.0. W3C Recommendation 27 June 2001 . 2001. Available: http://www.w3.org/TR/2000/PR-xlink-20001220/

[79] DeRose, S., Maler, E., and Daniel, R. Jr. XML Pointer Language (XPointer) Version 1.0. W3C Last Call Working Draft 8 January 2001 . 2001. Available: http://www.w3.org/TR/2001/WD-xptr-20010108

[80] Adler, S., Berglund, A., Caruso, J., Deach, S., Grosso, P., Gutentag, E., Milowski, A., Parnell, S., Richman, J., and Zilles, S. Extensible Stylesheet Language (XSL) Version 1.0. W3C Candidate Recommendation 21 November 2000 . 2000. Available: http://www.w3.org/TR/2000/CR-xsl-20001121/

[81] Clark, J. XSL Transformations (XSLT) Version 1.0. W3C Recommendation 16 November 1999 . 1999. Available: http://www.w3.org/TR/1999/REC-xslt-19991116

[82] Clark, J. and DeRose, S. XML Path Language (XPath) Version 1.0. W3C Recommendation 16 November 1999 . 1999. Available: http://www.w3.org/TR/1999/REC-xpath-19991116

[83] Lynch, C. Identifiers and their role in networked information applications. Bulletin of the American Society for Information Science 24[2], 17-20. 1997.

[84] Paskin, N. Toward unique identifiers. Proceedings of the IEEE 87[7], 1208-1227. 1999.

[85] Sollins, K. and Manister, L. Functional Requirements for Uniform Resource Names. RFC 1737 . 1994. Available: http://www.ietf.org/rfc/rfc1737.txt

[86] Arms, W. Y., Bianchi, C., and Overly, E. A. An architecture for information in digital libraries. D-Lib Magazine, February 1997 . 1997. Available: http://www.dlib.org/dlib/february97/cnri/02arms1.html

[87] Paskin, N. The DOI® Handbook version 2.3.0. doi.org . 2002. Available: http://www.doi.org/handbook_2000/index.html

[88] Moats, R. URN Syntax. RFC 2141 . 1997. Available: http://www.ietf.org/rfc/rfc2141.txt

[89] Daigle, L., van Gulik, D., Iannella, R., and Faltstrom, P. URN Namespace Definition Mechanisms. Internet-Draft . 2002. Available: http://www.ietf.org/internet-drafts/draft-ietf-urn-rfc2611bis-04.txt

[90] International Federation of Library Associations. Digital libraries: Metadata resources. IFLANET Electronic Collections . 2000. Available: http://www.ifla.org/II/metadata.htm

[91] Caplan, P. You call it corn, we call it syntax-independent metadata for document-like objects. The Public-Access Computer Systems Review 6 4. 1995. Available: http://info.lib.uh.edu/pr/v6/n4/capl6n4.html

[92] Gradmann, S. Cataloguing vs. Metadata: old wine in new bottles? 1998. 64th IFLA General Conference August 16 - August 21, 1998. 1998.

[93] Milstead , J. and Feldman, S. Metadata: Cataloging by any other name ... ONLINE, January 1999 . 1999. Available: http://www.onlineinc.com/onlinemag/OL1999/milstead1.html

[94] Chepesiuk, R., "Organizing the Internet: The "core" of the challenge," *American Libraries*, vol. 30, no. 1 (Jan 1999), pp. 60-63, 1999.

[95] Weibel, S. Metadata: The foundations of resource description. D-Lib Magazine, July 1995 . 1995. Available: http://www.dlib.org/dlib/July95/07weibel.html

[96] Dublin Core Metadata Initiative. Dublin Core Qualifiers. Dublin Core Metadata Initiative Recommendation . 2000. Available: http://purl.org/dc/documents/rec/dcmes-qualifiers-20000711.htm

[97] Lagoze, C. The Warwick Framework: A container architecture for diverse sets of metadata. D-Lib Magazine, July/August 1996 . 1996. Available: http://www.dlib.org/dlib/july96/lagoze/07lagoze.html

[98] Lassila , O. Web metadata: a matter of semantics. IEEE Internet Computing 2[4], 30-37. 1998.

[99] McGuinness, D. L., "Ontologies come of age," in Fensel, D., Hendler, J., Lieberman, H., and Wahlster, W. (eds.) *To appear in The Semantic Web: Why, What, and How* MIT Press, 2001.

[100] van Harmelen, F. OIL ontology inference and interchange. Presentation at 14th European Conference on Artificial Intelligence ECAI-00 . 2000. On-To-Knowledge project. Available: http://www.ontoknowledge.org/oil/presentations/index.shtml#ECAI00

[101] Heflin, J. and Hendler, J. Semantic interoperability on the Web. 111-120. 2000. Graphic Communications Association. Proceedings of Extreme Markup Languages 2000.

[102] Broekstra, J., Klein, M., Decker, S., Fensel, D., and Horrocks, I. Adding formal semantics to the Web: Building on top of RDF Schema. 2000. Lisbon, Portugal. Proceedings of the Workshop "ECDL 2000 Workshop on the Semantic Web". 9-21-2000.

[103] Heflin, J., Hendler, J., and Luke, S. SHOE: A knowledge representation language for Internet applications. Technical Report, CS-TR-4078 (UMIACS TR-99-71). 1999. Dept. of Computer Science, University of Maryland at College Park.

[104] Luke, S., Specter, L., and Rager, D. Ontology-based knowledge discovery on the World Wide Web. Franz, A. and Kitano, H. 96-102. 1996. AAAI Press. Working Notes of the Workshop on Internet-Based Information Systems at the 13th National Conference on Artificial Intelligence (AAAI96).

[105] Luke, S. and Heflin, J. SHOE 1.01 Specification. SHOE Project . 2000. Available: http://www.cs.umd.edu/projects/plus/SHOE/spec.html

[106] Heflin, J., Hendler, J., and Luke, S. Coping with changing ontologies in a distributed environment. 1999. AAAI-99 Workshop on Ontology Management.

[107] Heflin, J. and Hendler, J. Dynamic ontologies on the Web. 443-449. 2000. Menlo Park, CA, MIT-AAAI Press. Proceedings of 17th National Conference on Artificial Intelligence (AAAI-2000).

[108] Decker, S., Fensel, D., van Harmelen, F., Horrocks, I., Melnik, S., Klein, M., and Broekstra, J. Knowledge Representation on the Web. 2000. Aachen, Germany. Proceedings of the 2000 International Workshop on Description Logics (DL2000). 2000.

[109] Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D. L., and Patel-Schneider, P., "OIL: An ontology infrastructure for the Semantic Web," *IEEE Intelligent Systems*, vol. 16, no. 2, pp. 38-45, 2001.

[110] Horrocks, I., Fensel, D., Broekstra, J., Decker, S., Erdmann, M., Goble, C., van Harmelen, F., Klein, M., Stabb, S., Studer, R., and Motta, E. The Ontology Inference Layer OIL. Technical Report. 2000. University of Manchester / Vrije Universiteit Amsterdam.

[111] Horrocks, I. A denotational semantics for Standard-OIL and Instance-OIL. OIL Homepage . 11-29-2000. Available: http://www.ontoknowledge.org/oil/downl/semantics.pdf

[112] Horrocks, I. and Sattler, U. Ontology reasoning for the semantic web. 2001. Proceedings of the 17th Int.Joint Conf.on Artificial Intelligence (IJCAI'01).

[113] Dean, M. DARPA Agent Markup Language (DAML) update. Presentation at CoABS Principle Investigater Meeting . 2-2-2001. Available: http://www.daml.org/2001/02/coabs-daml/Overview.html

[114] Stein, L. A., Connolly, D., and McGuinness, D. L. DAML-ONT initial release. DAML Home Page . 10-1-2000. Available: http://www.daml.org/2000/10/daml-ont.html

[115] Horrocks, I. and van Harmelen, F. DAML+OIL (December 2000). DAML Home Page . 2000. Available: http://www.daml.org/2000/12/daml+oil-index.html

[116] Horrocks, I., van Harmelen, F., and Patel-Schneider, P. DAML+OIL (March 2001). DAML Home Page . 2001. Available: http://www.daml.org/2001/03/daml+oil-index.html

[117] Horrocks, I., van Harmelen, F., and Patel-Schneider, P. DAML+OIL (March 2001): A Datatype Extension to DAML+OIL (December 2000). DAML Home Page . 2001. Available: http://www.daml.org/2001/03/differences-daml+oil.html

[118] van Harmelen, F., Patel-Schneider, P., and Horrocks, I. Reference description of the DAML+OIL (March 2001) ontology markup language. DAML Home Page . 2001. Available: http://www.daml.org/2001/03/reference.html

[119] McGuinness, D. L. and van Harmelen, F. OWL Web Ontology Language Overview. W3C Recommendation 10 February 2004 . 2004. Available: http://www.w3.org/TR/owl-features/

[120] Palmer, S. Documents, Cars, Hills, and Valleys. www-rdf-interest@w3.org. 2002. 9-4-2002.

[121] Daconta, M. C. Associations in RDF. www-rdf-interest@w3.org. 2002. 7-17-2002.

[122] Hewlett-Packard Company. The jena semantic web toolkit. HP Laboratory Semantic Web activity . 2002. Available: http://www.hpl.hp.com/semweb/jena-top.html

[123] Horrocks, I. and Tessaris, S. Querying the semantic web: a formal approach. 2002. Springer-Verlag. Proceedings of the 2002 International Semantic Web Conference (ISWC 2002). Horrocks, I. and Hendler, James.

[124] Haarslev, V. and Möller, R. Description of the RACER system and its applications. 2001. Proceedings of the 2001 International Description Logics Workshop (DL-2001). Goble, C. A., McGuinness, D. L., Möller, Ralph, and Patel-Schneider, Peter.

[125] Tessaris, Sergio, "Questions and answers: reasoning and querying in Description Logic." Ph.D University of Manchester, 2001.

[126] Patel-Schneider, P. and Swartout, B. Description Logic Knowledge Representation System Specification from the KRSS Group of the ARPA Knowledge Sharing Effort. the KRSS Group of the ARPA Knowledge Sharing Effort . 1993. Available: http://www-db.research.bell-labs.com/user/pfps/papers/krss-spec.ps

[127] Haarslev, V. and Möller, R. RACER User's Guide and Reference Manual version 1.6.1. RACER Homepage . 2001. Available: http://kogs-www.informatik.uni-hamburg.de/~race/racer-manual-1-6-1.pdf

[128] Horrocks, I. DAML+OIL: a Description Logic for the Semantic Web. IEEE Data Engineering Bulletin 25[1], 4-9. 2002.

[129] Fleiss, J. L., *Statistical Methods for Rates and Proportions*, 2 ed. John Wiley & Sons, 1981.

[130] Newcombe, R. G., "Two-Sided Confidence Intervals for the Single Proportion: Comparison of Seven Methods," *Statistics in Medicine*, vol. 17 pp. 857-872, 1998.

[131] van Rijsbergen, C. J., "Evaluation," *Information Retrieval* 2 ed. London: Butterworths, 1979.