Hao Ding

A Semantic Search Framework in Peer-to-Peer Based Digital Libraries

Department of Computer and Information Science
Norwegian University of Science and Technology
N-7491 Trondheim, Norway

NTNU Norwegian University of Science and Technology

Doctoral thesis
for the degree of Doktoringeniør

Department of Computer and Information Science
Faculty of Information Technology, Mathematics and Electrical Engineering

ISBN 82-471-8154-1 (printed version)
ISBN 82-471-8153-3 (electronic version)
ISSN 1503-8181

Doctoral theses at NTNU, 2006:190

Printed by NTNU-trykk

To MY LOVING WIFE YUN LIN
AND OUR DAUGHTER SABRINA DING

Abstract

Advances in peer-to-peer overlay networks and Semantic Web technology will have a substantial influence on the design and implementation of future digital libraries. However, it remains unclear how best to combine their advantages in constructing digital library systems. This thesis is devoted for investigating, proposing and evaluating possible solutions to advance developments in this field.

The main research goal of this work is to combine the strengths of both peer-to-peer overlay networks and Semantic Web for facilitating semantic searches in large-scale distributed digital library systems. The approach has been conducted in a sequential and progressive manner. Firstly, we recognize system infrastructure and metadata heterogeneity as two major challenges in conducting semantic searching across distributed digital libraries. Next, we investigate the strengths and weaknesses of both peer-to-peer and Semantic Web technology and justify that these two fields are complementary and can be combined in conducting semantic searches in a large-scale distributed environment. Thirdly, due to various topologies, functionalities and limitations different peer-to-peer infrastructures may possess, we survey current classical peer-to-peer systems so as to facilitate determinating appropriate infrastructure for specific application scenario. Fourthly, we probe into approaches in generating ontology-enriched metadata records for semantic search purpose. Finally and most importantly, we will propose a semantic search process for interoperation among heterogeneous resources, basing on ontology mapping mechanism.

A major contribution expected in our work is, in a broader term, proposing and investigating possible solutions in combining the strengths of both peer-to-peer overlay networks and Semantic Web for facilitating semantic search among highly distributed digital libraries. From a specific perspective, we provide an appropriate benchmark for facilitating decision making in choosing appropriate peer-to-peer networks for digital library construction; especially, we consider in this work no global schema exists and further justify the feasibility and advantages of ontology engineering method in semantic enriched metadata management; to support federated search in such a distributed environment, we also propose an extended super-peer network model, emphasizing in load-balancing and self-organizing capabilities; Based on semantic enriched metadata management, we propose also direct ontology mapping method to enable runtime semantic search process. Evaluation results have illustrated the feasibility

and robustness of our approaches.

The future direction of this work includes studies on user authentication, efficient ontology parsing and real-life applications.

Contents

Preface **xv**

1 Introduction **1**
1.1 Motivation 1
1.2 Research Questions 4
1.3 Objectives 6
1.4 Approach 7
1.5 Major Contributions 8
1.6 Thesis Outline 9

2 Digital Libraries **11**
2.1 Defining Digital Library 11
2.2 Concepts in Digital Library 13
2.2.1 Digital Object 14
2.2.2 Collection 14
2.2.3 Metadata 15
2.2.4 Identity 19
2.3 Instances of Metadata Standards 21
2.3.1 MARC 21
2.3.2 Dublin Core 21
2.4 Search Protocols in Digital Libraries 23
2.4.1 Z39.50 24
2.4.2 The Open Archive Initiative 25
2.4.3 Dienst 26
2.4.4 Simple Digital Library Interoperability Protocol (SDLIP) 27
2.5 Information Searching in Digital Libraries 28
2.5.1 Keyword-based Information Retrieval 28
2.5.2 Metadata-based Search 29

2.5.3 Ontology-based Search 30
2.5.4 Federated Search Solutions 32
2.6 Challenges in Distributed Information Search in Digital Libraries 35
2.7 Chapter Achievement 36

3 P2P Overlay Network and Digital Libraries 37
3.1 Introduction 37
3.2 Why P2P in Digital Libraries? 37
3.2.1 Digital Library Architecture of the Past and Current 37
3.2.2 Digital Library Architecture of the Future 40
3.3 Various P2P Models 41
3.3.1 Pure P2P Model 41
3.3.2 Hybrid P2P Models 42
3.3.3 Super-Peer based P2P Model 44
3.4 Existing P2P Systems 45
3.4.1 Gnutella, Naspter and Freenet 45
3.4.2 Routing Indices 45
3.4.3 Distributed Hash Table (DHT) 46
3.4.4 P-Grid 46
3.4.5 HyperCup 47
3.4.6 Piazza 47
3.4.7 JXTA Search Edutella, Bibster 48
3.4.8 RDFPeers 48
3.4.9 OAI-P2P 48
3.4.10 Semantic Overlay Networks 49
3.5 Challenges in P2P Networks 49
3.6 Chapter Achievement 50

4 Appropriate P2P Infrastructures for Semantic Search 53
4.1 Introduction 53
4.2 Benchmark for Selecting Appropriate P2P Architectures . . 53
4.3 A Super-Peer based Network Supporting Federated Sesarch 57
4.3.1 Requirements 57
4.3.2 Classic Super-Peer System Model - Revisited 59
4.3.3 Enhanced Super-Peer Model for Federated Search . 60
4.4 Evaluation 62
4.4.1 Evaluation Setting 62

4.4.2 Experiment 1 - Super-peer Network Generation via Gossiping protocol . . . 64
4.4.3 Experiment 2 - Load Balancing . . . 66
4.4.4 Experiment 3 - Self Organizing . . . 66
4.4.5 Summary . . . 68
4.5 Chapter Achievement . . . 70

5 Metadata Heterogeneity 71
5.1 Introduction . . . 71
5.2 Heterogeneity Categories . . . 71
5.2.1 Integrating Syntactically Heterogeneous Sources . . . 72
5.2.2 Integrating Structurally(Schematically) Heterogeneous Sources . . . 72
5.2.3 Integrating Semantically Heterogeneous Sources . . . 75
5.2.4 Metadata Encoding Methods . . . 80
5.3 The Needs for Explicit Semantics . . . 82
5.4 Chapter Achievement . . . 84

6 The Semantic Web and Digital Libraries 85
6.1 Introduction . . . 85
6.2 Digital Libraries Challenges - Revisited . . . 86
6.3 The Semantic Web . . . 88
6.3.1 Brief History of the Semantic Web . . . 88
6.3.2 Ontology - The Key Enabler for the Semantic Web . . . 88
6.3.3 The Semantic Web Languages . . . 90
6.4 Description Logics and Ontologies . . . 93
6.4.1 Formal Syntax and Semantic of Description Logics . . . 94
6.4.2 Description Logics-based Reasoning . . . 96
6.5 Inferential Rule based Ontology Translation . . . 98
6.6 The Importance of Applying Semantic Web in Digital Libraries . . . 99
6.7 Chapter Achievement . . . 99

7 Semantic Enriched Metadata Management 101
7.1 Introduction . . . 101
7.2 The Role of Metadata, Context and Ontologies . . . 102
7.3 Developing Ontological Knowledge Sources . . . 103
7.3.1 Ontology Creation or Import . . . 104
7.3.2 Enriching Metadata Records with Semantics . . . 106

7.3.3 Semantic Information Search . . . 108
7.3.4 Semantic Information Usage . . . 108
7.4 Interoperating Semantically Heterogeneous Sources . . . 109
7.4.1 Current Approaches . . . 109
7.4.2 Overview on Semantic Interoperation Methods . . . 112
7.5 Chapter Achievement . . . 114

8 Semantic Relations Elicitation 117
8.1 A Process for Enabling Semantic Search in P2P Network . 117
8.2 General Definitions and Hypotheses . . . 119
8.2.1 Meanings of 'Run-time' . . . 119
8.2.2 Peer Communication Model . . . 119
8.2.3 Research Hypotheses . . . 120
8.3 Semantic Elicitation . . . 120
8.4 Semantic Bridges Between Concepts . . . 121
8.5 Semantic Elicitation via Logic Reasoning . . . 123
8.5.1 Requirements for Describing Complex Relations . . 123
8.5.2 Combining Ontologies with Rules . . . 125
8.6 Walk-Through Examples . . . 128
8.7 Evaluation . . . 131
8.7.1 Evaluation Settings . . . 132
8.7.2 Evaluation Results . . . 135
8.8 Chapter Achievement . . . 137

9 Prototype Implementation 139
9.1 Prototype Architecture . . . 139
9.2 UML Diagram of Prototype . . . 142
9.3 Adopted Technologies . . . 143
9.3.1 JXTA Framework . . . 143
9.3.2 SESAME . . . 144
9.3.3 KAON2 . . . 146
9.4 GUI . . . 146
9.5 Chapter Achievement . . . 148

10 Conclusion 149
10.1 Answers to the Research Questions . . . 149
10.2 Contributions . . . 150
10.3 Limitations and Future Work . . . 151

A List of Publications **153**

List of Figures

1.1 Combining Semantic Web and P2P for Digital Libraries . . 4

2.1 The Multiple Purposes of Using Metadata. 16
2.2 A MARC record (partial), generated from BIBSYS[1] . . . 22
2.3 The Generic Retrieval Process. Adapted from [2] (pp.10) . 29
2.4 A Simple Metadata Creation and Search Process. 30
2.5 A Simple Ontology-based Search Process. 32
2.6 An Ontology-based Search Process Powered by Rule Inferencing. 33
2.7 Federated Search Diagram, from [3] 34

3.1 The Ad-hoc Digital Library Architecture 38
3.2 A Middleware-based Digital Library Architecture 39
3.3 The Pure Peer-to-Peer Model. 42
3.4 The P2P with Simple Discovery Server Model. Only the discovery of clients occurs via the server; the rest of the communication occurs among peers 43
3.5 P2P with a Discovery, Lookup, and Content Server 44
3.6 Super-Peer based P2P Model 45
3.7 Hypercube Graph and Serialization Notation, from [4] . . . 47

4.1 Super-Peer Network Generated by Gossiping Protocol(Outdegree) 64
4.2 Super-Peer Network Generated by Gossiping Protocol (Capacity) 65
4.3 Load Balancing (1000 Nodes join the network each time from round 25 to round 35) 66
4.4 Self-Organizing in Scenario of Continuous Peer Leaving . . 68
4.5 Catastrophe Recovery (50000 Nodes left in round 5) 69

5.1 A GAV Example 73
5.2 A LAV Example 74
5.3 A LAV Example 75

6.1 Example: University Taxonomy and Ontology 89
6.2 RDF Data Model 91
6.3 An RDFS ontology 92
6.4 Architecture of a Knowledge Representation System based on Description Logics. From [5] (pp.50) 96
6.5 Semantic Web Track. From [6] 98

7.1 The Relations Among *Context*, *Metadata* and *Ontology*. Adapted from [7]. 102
7.2 Using RDF(S) to Represent Relations between Metadata Terms 103
7.3 The Development Process for Ontological Knowledge Sources. 104
7.4 The General Ontology Creation Process. 104
7.5 RDF Data Model 107
7.6 The Most Often Applied Methods in Inter-relating Ontologies 109
7.7 The Continuum of Semantic Interoperation Methods . . . 112

8.1 The Mapping Process 118
8.2 The Continuum of Semantic Interoperation Methods 119
8.3 Portions of two bibliographic ontologies 125
8.4 Processing Complex Relations 126
8.5 Configuration File 134
8.6 Accuracy 135
8.7 Execution Time 136

9.1 General Peer Architecture 140
9.2 System Communication Layer 141
9.3 Upper Leverl Class Diagram (Interfaces) 142
9.4 Peer Discovering and Joining 144
9.5 The SESAME architecture, from [8] 145
9.6 The Prototype GUI 147
9.7 The Manual Mapping GUI 147

List of Tables

2.1 Summary of Dublin Core. Further details from [9] 23
2.2 Facilities in Z39.50. From [10] pp.428 24
2.3 OAI Protocol Requests 25

3.1 Advantages and Disadvantages of P2P Systems (Compared with Client/Server Systems) 50

4.1 Summary of Typical P2P Systems. From [11] 55
4.2 Working Benchmark for Selecting P2P Infrastructures for Digital Libraries 58
4.3 Configuration parameters and default values 63

5.1 MARC to unqulaified Dublin Core Crosswalk, from [12] . . 77

6.1 Syntax and Semantics of Description Logic Constructors . . 95

7.1 Comparison of Semantic Interoperation Approaches in P2P Setting 115

8.1 The Resulting Knowledge Base in $< \mathcal{O}_s, \mathcal{O}_t, \mathcal{M}_{s,t} >$ 129
8.2 Two ontologies in 'Time' 133
8.3 Two ontologies in 'Bibliography' 133
8.4 Two ontologies in 'Tourism' 134

List of Algorithms

1 2D Mapping Table Generation Algorithms 77

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) for the degree *"doktor ingeniør"*. The work has been carried out during the time period 2002-2006 at the Information Management Group, Department of Computer and Information Science.

Acknowledgments

First and foremost, I would like to thank my supervisor — Prof. Ingeborg Torvik Sølvberg. It is no more than lucky to have her to guide me through my Ph.D research. She is not only knowledgeable but precise, always being able to find my problems in the first time and teach me how to correct them. Under the supervision of my supervisor, I have learned "how to work the plan, and plan the work". I deeply believe that I can benefit from it in my whole life.

I am grateful to Prof. Jon Atle Gulla and Ass. Prof. Trond Aalberg. They have spent a lot of time in reviewing my thesis, informing me the latent problems and advising me the right methods in conducting research.

Øyvind Vestavik has been there to support me through the good times and the bad. I thank him for his patience and his wit, for always listening, and for inspiring me in ways he probably does not even realize. I could not ask for a better friend and partner. Thank Peep Kügas for all the interesting discussions on "super-peer", Semantic Web and distributed computing. Thank Janny, Jeanie, Lars and Christian for all the helps, talks and discussions.

Thanks to all at IDI, in particular my colleagues in the Information System Group, for the cooperative and stimulating working atmosphere. Thank Will Young and Celine Dion for their beautiful music I have listened to when writing the thesis.

This thesis is dedicated to my family who have always provided me with the highest degree of love and support. This includes my parents (Jianhua Ding and Huanzhen Wang) as well as my elder sister and brother-in-law (Yan Ding and Jianbin Wan), my loving wife (Yun Lin) and our feline daughter (Sabrina Ding).

Hao Ding
October 11, 2006

Chapter 1

Introduction

The topic of this thesis is to investigate how to best combine Peer-to-Peer(P2P) and Semantic Web technologies for semantic searching across largely distributed and heterogeneous digital libraries. The major research tasks involved are to apply appropriate infrastructure for specific digital library system construction, to enrich metadata records with ontologies and enable semantic searching upon such system infrastructure. The "semantic search" in this thesis is a specialized functionality that discovers, analyzes and interoperate semantically related metadata records dispersed in different collections in a digital library system. In this chapter, we are to present the motivation and objectives of our work, the research questions and contributions, and as well as the organization of this thesis.

1.1 Motivation

The motivation behind this research comes in a broad way from the recognized tendency of building the digital library systems of the future allowing users to access collections in various forms at any time, from anywhere, and in an efficient and effective way. Due to that the number of publicly available digital libraries is increasing sharply and they are in fact managed by many independent organizations on different topics and for different user preferences, few libraries have more than a small percentage of the collections that users might want. Therefore, one significant effect is that users have to explore different collections and services for appropriate information.

Many approaches have been conducted to weave mutually interested

digital libraries together in a coherent way so as to provide a 'one-stop' service for users. For example, federated digital libraries [13, 14] for providing interoperability among their members; union catalogs for exposing library holdings to be openly accessible; and Simple Digital Library Interoperability Protocol (SDLIP) in Stanford which works as a 'search middleware' for integrating heterogeneous digital libraries. These solutions can be regarded as a centralized solution to some extent, because they need a centralized server to administrate or organize participating libraries.

However, such client/server architectures come along with some constraints [15] as well. First, all clients depend heavily on servers. That is, servers are responsible for a centralized control of whole system. Such an architecture may not scale well since servers may easily become bottleneck when too many user queries flood in. Second, many libraries are in practice independent or at least loosely coupled. That is, a degree of *autonomy* is required in such systems, which may then not be able to join such a federated network. It is easy to understand this point since their first duty is to cater for the users in the local community and sharing their resources with other communities will always come next.

As opposed to the client/server architecture, a more dynamic and scalable architecture, Peer-to-Peer (P2P) overlay network, becomes an alternative to reduce the dependency on the server and the centralization of control from servers. P2P-based architecture holds many promises over client/server architecture and alleviates the aforementioned problems somehow. In a P2P system, nodes typically connect to a small set of random nodes (their neighbors) in order to fulfill a task, such as file searching or service discovery. Consequently, it can scale up easily at user's will. It also alleviates the bottleneck problem since P2P systems can work without any centralized server at all. In addition, P2P network helps increase system accessibility, such as distributing query processing tasks to multiple computing nodes. Hence, a study over various kinds of P2P architecture models is worthwhile for how and to what extent they can be integrated into digital library system constructions.

The P2P infrastructure solves *partially* our intention to access digital libraries freely, but it does not deal with any *heterogeneity* issues in library collections dispersed in such a distributed environment. Firstly and clearly, it is impractical for library developers to force all participating libraries conform to certain *standards* in syntax, structure, and semantics, because of diversities in application profiles, users' requirements, types of collections and depth of descriptions. Basically, the spectrum of *hetero-*

geneity, according to Ouksel and Sheth [16], may exist in system infrastructure, abstract syntax, information structure and semantics. Syntactic and structural heterogeneities are encompassing issues in handling, exchange and combining of metadata records properly, with special regard to formats, encodings, properties, data types and so forth [17]. Many sophisticated solutions to syntactic and structural issues have been well reported (c.f. [18, 19, 20, 21, 22, 23]) although there are still some difficulties in association with the structure of terminology as stressed in [24]. So, throughout this thesis we will mainly focus on the issue of semantic interoperability which is significant and inevitable if we want to move on to solve all heterogeneity problems. In our context, discrepancies in semantics, or simply semantic heterogeneities result from the semantic conflicts in terms, phrases, and context, which are adapted in different metadata schemas expressed in various ways. In order to explicate the contents, essential properties, and relationships between metadata elements, a widely accepted method is to apply the Semantic Web [25] and *Ontology* [26] technologies. The Semantic Web technologies brought forward by Tim Berners-Lee, have opened up knowledges in Web pages by enriching their content with semantic meta-information that can be processed by inference-enable applications. Ontologies are used to clarify relations between ambiguous concepts so as to discover common meanings shared by different documents. One simple example is interrelating two metadata terms author and creator used to annotate in bibliographic records. If referring to Dublin Core [9], author should be subsumed by creator.

Currently, empowering ontologies with rule inferencing functionality has also become a hot spot in the Semantic Web research [27]. In fact, *inferential rules* are a major issue in further developing intelligent applications in the Semantic Web (c.f. [28]). On one hand, they can be used in ontology languages, either in conjunction with or as an alternative to description logics. On the other hand, they will act as a means to draw inferences, to express constraints, to specify policies, to react to events/changes and to transform data. In practical digital library applications, although there are approaches adopting ontology-based approach to enhance precision in accessing local specific collections [29, 30, 31], little research has been conducted so far in interoperability between heterogeneous metadata schemas so as to facilitate searching in distributed environment.

In summary, the P2P overlay network and the Semantic Web technologies are to have substantial effects on design and implementation of

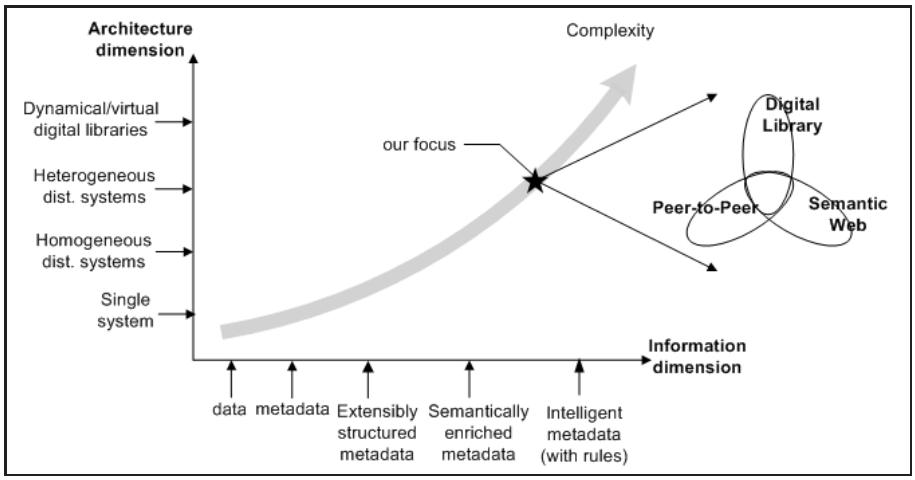

Figure 1.1: Combining Semantic Web and P2P for Digital Libraries

future digital libraries. Technological advances in these areas have made several new possibilities and challenges apparent. It also triggers the motivations for the work we are to present in this thesis. We illustrate our standing point in the holistic research picture in Figure 1.1. In a more specific way, that is: applying appropriate P2P overlay network to obtain a new communication infrastructure for interlinking autonomous digital library systems; using ontologies, instead of simply metadata mapping to process terminological relationships between terms across ontologies; and additionally adopting inferential rules to explicate more complex semantic relations between different ontologies. Such integrated approach is expected to result in an articulation or linkage between library systems from both the systematic and semantic perspectives which further facilitate searching across federated digital libraries.

1.2 Research Questions

With respect to the listed scientific challenges above, the main research question that this thesis attempts to answer is: *How and to what extent a P2P architecture extended with semantic technologies can enable search of the same quality as if the system was one centralized library?* Challenges related to this question may involve various issues, such as resource discovery, information storage, organization and searching, and interoper-

ability in digital libraries [32]. In order to narrow the research scope, we have specified and decomposed the research question into three subtasks as follows:

- *How suitable are various P2P infrastructures in decentralized digital library solutions?* As mentioned previously, quite a few P2P overlay networks have emerged in various applications, covering different application domains (e.g. file sharing, distributed computing and distributed search), having different requirements (e.g. autonomy) and embodying different functionalities(e.g. simple keyword-based search, semantic search). Although some P2P-based applications have appeared in digital library community [33] [34] [35], there is not a benchmark or guideline for digital library developers to determine suitable P2P infrastructure for their specific usages. Often a P2P infrastructure successfully implemented in one digital library system might not be suitable in others. Therefore, it is necessary to compare typical P2P infrastructures and exploit their adaptabilities for digital library applications.

- *What kind of metadata interoperation/integration method should be adopted in P2P-based digital library systems?* When conducting search over P2P-based digital libraries, it is normal to have libraries created in distinct or overlapping metadata schemas. It is thus difficult to access corresponding collections in these libraries even if they are open for incoming queries. Hence, an extra step of metadata mapping is necessary for bridging relations between relevant metadata elements. Generally, there can be two branches of approaches for integrating heterogeneous information, namely, the global schema based and direct mapping based approaches. From the theoretic perspective, it is possible to apply both of these methods in P2P-based libraries. However, which method is more suitable in what kind of situation in P2P-based communication is not very clear. In this thesis, we are to investigate this issue from several perspectives, such as frequencies of peer's joining/leaving in P2P systems, availability of cache, documents' popularity and P2P typologies which may be adopted in different situations.

- *How suitable semantic technologies (ie. ontologies and inference mechanisms) are in elicitating implicit semantic relations between schemas and supporting search?* Recent approaches seem fairly

promising in automatically deducing relations between relevant records via description logics-based [5] ontology languages, such as OIL, DAML+OIL, and OWL. Such an approach has some problems in expressing more complex and important relations, eg. user defined rules. More advanced approaches, such as logic-based reasoning, can also be adopted to establish correspondences between concepts/properties. If one could have an inference engine seamlessly integrated with normal ontology languages, possibly more implicit relations may be derived. However, it remains unclear whether it is worth to integrate inferential rules in P2P-based systems where rule inferencing is expensive to conduct. In this thesis, we are to study the cost and benefit of adopting ontologies and inference mechanisms in supporting search.

Beyond these, there are also many other critical but less relevant research questions, such as storage and query processing. These topics will not be covered in this thesis although they are often highly concerned in building P2P systems.

1.3 Objectives

The research goal of this work is how semantic technologies and P2P infrastructures can solve the problems in distributed digital libraries from the search perspective. Note that the focus of our work is to *enable* search, instead of the search *process* itself. In this thesis we assume that a standard search engine is available and query reformulations will depend heavily on such search engine. More specifically, the objectives are decomposed into:

- explore and understand the requirements for rendering semantic search in P2P networks;
- investigate from a search perspective possible P2P infrastructures for constructing decentralized digital libraries where no global schema exists;
- investigate how the semantic technologies can be used for elicitating additional semantics from existing resources;
- analyze the implementation results, and evaluate the feasibility of our approaches in enabling search in P2P-based digital libraries.

Among the above listed objectives, the key focus of this work is for enabling searching across heterogeneous metadata records dispersed in the P2P network. Given the immature nature of the P2P network, a considerable work is needed to survey the strengths and weaknesses of current available P2P systems. Additionally, a special mechanism is required to handle interoperation between heterogeneous metadata schemas so as to interlink dispersed records together.

1.4 Approach

An overview of the research activities in this thesis is as follows.

- *Extensive review on the Semantic Web and P2P technologies:*
  Due to the multi-disciplinary nature of this work (i.e., digital library, P2P and the Semantic Web), the spectrum of literature is rather wide. After narrowing the scope, we focus on metadata interoperability, P2P architecture model and ontology engineering. A clear understanding is required on the strengths and weaknesses of the Semantic Web and P2P technologies.

- *Proposing benchmarks for choosing appropriate P2P infrastructures for specific digital library applications:*
  it includes investigating typical current P2P systems and corresponding infrastructures. Descriptive analyses on these systems should be conducted and special considerations on their adaptabilities to distributed digital library construction will be investigated.

- *Researches on metadata integration strategies:*
  it includes the investigation of related works and approaches in metadata interoperation. In our setting, special consideration is on how to cope with the dynamic nature of P2P-based communication where there is no central management, or administration. In other words, our work focuses on generic domain instead of specific ones.

- *A general framework for generating semantically enriched resources:*
  it consists of applying the Semantic Web technologies to evolve conventional metadata mapping to ontological knowledge level mapping and interoperation. Particularly, rule-based logic reasoning will be studied in explicating implicit relationships among heterogeneous resources.

- *Prototype design and implementation:*
  it concerns the design of a prototype system for semantic search framework, in order to verify that our proposed approach is an applicable solution. Note that prototype implementation is for proof-of-concept purpose, rather than fulfilling all functionalities mentioned in this thesis.

- *Evaluation:*
  It includes evaluating tentative P2P overlay networks in term of their applicability in dynamic and scalable digital libraries. In addition, evaluation is also needed for testing ontology mapping method which is critical in achieving run-time semantic search.

1.5 Major Contributions

A major contribution expected in our work is, in a broader term, proposing and investigating possible solutions in combining the strengths of both peer-to-peer overlay networks and Semantic Web for facilitating semantic searches in large-scale distributed digital libraries. Specifically, the following contributions have been achieved in this thesis:

1. Providing appropriate benchmarks for facilitating decision making in choosing appropriate peer-to-peer networks for digital library construction. To support federated search in such a distributed environment, we also propose an extended super-peer network model, emphasizing in load-balancing and self-organizing capabilities.

2. Proposing a P2P-based semantic search framework where *no* global schema exists. We further justify the feasibility and advantages of ontology engineering method in semantic enriched metadata management.

3. Applying ontology engineering methodology in metadata integration and developing a ontology mapping component for this work. Evaluation results justify that our design and mechanism can achieve reasonable response time with satisfactory precision.

4. Developing a prototype application for enabling semantic search over distributed digital libraries.

1.6 Thesis Outline

The remaining of this thesis is laid out as follows:

- **Chapter 2** introduces the major characteristics of digital library systems and commonly used search protocols and methods in digital libraries. This chapter further discusses challenges in conducting distributed information search in digital libraries, mainly in aspects of system infrastructure and semantic interoperability.

- **Chapter 3** introduces general P2P models and typical P2P systems and discusses the reasons why research in P2P networks is important to digital library community. Advantages and disadvantages of P2P networks will be presented in this chapters as well.

- **Chapter 4** summarizes P2P system models and proposes a benchmark, helping determining appropriate P2P infrastructures for digital library constructions under specific requirements. An enhanced super-peer model will be proposed for supporting our federated searching requirement. Evaluations are to be conducted to justify the proposals.

- **Chapter 5** presents current approaches in metadata interoperation methods in aspects of structure, syntax and semantics. The need for explicit semantic will be justified.

- **Chapter 6** revisits the challenges in digital libraries, introduces benefits of applying Semantic Web technologies for processing 'semantics' and justifies its importance in addressing semantic interoperation problems. Description Logics will be introduced, with a special intention for justifying that logic-based reasoning is useful in explicating complex relations.

- **Chapter 7** describes an abstract model for employing domain specific ontologies to bridge the heterogeneities in various metadata schemas. A general process is to be presented for creating ontological knowledge sources. Different approaches on ontology interoperation will be described and compared as well, especially in concern of dynamic P2P computing scenarios.

- **Chapter 8** presents a semantic search process for interoperation among heterogeneous ontologies in distributed environment - which

is the critical issue to conduct semantic search in P2P scenarios. Evaluation is to be conducted as well in this chapter.

- **Chapter 9** describes a prototype system, illustrating the feasibility of conducting semantic search in a super-peer based digital library. System architecture, components and functionalities will be presented as well in this chapter.

- **Chapter 10** concludes this work with a summary of major contributions, limitations and future work.

Chapter 2

Digital Libraries

2.1 Defining Digital Library

Digital Libraries (DL) have seen a significant increase in use over past several decades across multiple research domains. The term *digital libraries* itself has thus a variety of potential meanings, ranging from a digitized collection of material that one might find in a traditional library through to the collection of all digital information along with the services that make that information useful to all possible users (e.g. Internet search engines, library systems) [36]. Such a variety leads to many definitions, instead of a commonly accepted one, for digital libraries emphasizing different aspects. Waters [37] provides a working definition for digital libraries:

> *Digital libraries are organizations that provide the resources, including the specialized staff, to select, structure, offer intellectual access to, interpret, distribute, preserve the integrity of, and ensure the persistence over time of collections of digital works so that they are readily and economically available for use by a defined community or set of communities.*

In a more precise way, Borgman [38] describes DL as follows:

> *Digital libraries are a set of electronic resources and associated technical capabilities for creating, searching and using information. In this sense they are an extension and enhancement of information storage and retrieval systems that manipulate digital data in any medium (text, images, sounds; static or dynamic images) and exist in distributed networks. The content of digital libraries includes data, metadata that describe various aspects of the data (e.g., representation, creator, owner, reproduction rights), and metadata that consist of links or relationships to other data or metadata, whether internal or external to the digital library.*

Essentially, distinct views on digital libraries can be drawn out since they involve not only database and networking technologies, but also user preferences, languages and knowledge representation. In order to have a consistent form and clear conceptualization, we conform the definitions of "Digital Library", "Digital Library System", and "Digital Library Management System" to the Reference Model for Digital Library Management Systems [39].

Definition 2.1 *Digital Library (DL):A (potentially virtual) organization that comprehensively collects, manages, and preserves for the long term rich digital content and offers to its user communities specialized functionality on that content, of measurable quality, and according to prescribed policies.*

Definition 2.2 *Digital Library System (DLS): A software system that is based on a (potentially distributed) architecture and provides all functionality that is required by a particular Digital Library. Users interact with a Digital Library through the corresponding Digital Library System.*

Definition 2.3 *Digital Library System (DLMS): A generic software system that provides the appropriate software infrastructure to both (i) produce a basic Digital Library System that incorporates all functionality that is considered foundational for Digital Libraries and (ii) integrate additional software offering more refined, specialized, or advanced functionality. An intrinsic part of DLMS functionality is related to administrative services that are used to choose the appropriate subset of its functionality, e.g., through relevant parameters of its components, and then (potentially (semi-)automatically) install, deploy, and (re)configure a Digital Library System.*

We assume throughout this thesis that digital library systems exist in *distributed* environments. We then focus particularly on the problem of searching across distributed digital libraries using *heterogeneous* metadata schemas, which narrows our research scope to the management layer instead of the whole complicated framework. In this context, the following selected characteristics are important:

- Many digital libraries reside in a distributed and open environment into which individual libraries can be easily plugged. This may result in the ever-growing number of digital libraries and scalability will be supported in the entire digital library system.

- Each participating digital library is self-independent, which is different from the functionality of the ones in client/server infrastructure. In another word, all digital libraries in a cooperative system are essentially autonomous and can provide services to local users even if disconnected from such a system. Therefore, particular effort is needed to investigate specific networking infrastructure upon which largely independent digital libraries can cooperate.

- Relevant information can be found across multiple sources, such as a library catalog, a digital library repository and abstracting/indexing databases. Obviously, these collections can be physically isolated and thus queries have to be rendered over multiple sources.

- Given the non-monolithic nature of digital libraries, metadata must be correlated before answering heterogeneous requests. In addition to the approach to metadata interoperation per se, specific infrastructure should be built to facilitate both the development and operation of an interoperable system.

2.2 Concepts in Digital Library

The 'digital' characteristic of digital libraries increases the accessibility of the content to the user. Subject to the physical constraints of weight and distance in the traditional library, most physical items are organized in *collections*. These items are considered as *digital objects* which are generally described by *metadata* in digital libraries. *Identities* may also be used in metadata records so as to facilitate operations, such as acquiring, discovering and selecting digital objects. Throughout this thesis, we use also

the term *information resources* to indicate total means, such as metadata schemas and digital objects in digital library systems.

This section is devoted to introduce these basic but important concepts which will be used throughout this thesis, thus it is not trivial to clarify the corresponding meanings here.

2.2.1 Digital Object

For consistent and precise purpose, we use in this thesis the interpretation of *digital objects* in the Handle system [40].

*A **digital object** has a machine and platform independent structure that allows it to be identified, accessed and protected, as appropriate. A digital object may incorporate not only informational elements, i.e., a digitized version of a paper, movie or sound recording, but also the unique identifier of the digital object and other metadata about the digital object. The metadata may include restrictions on access to digital objects, notices of ownership, and identifiers for licensing agreements, if appropriate.*

2.2.2 Collection

In a certain sense, often libraries are regarded as a synonym of their *collections*. In fact, collection changes the ways of looking for information, selecting 'materials' and accessing materials. Generally, library collections serve four basic purposes [41]:

- *Preservation:* keeping materials for the future, as they may be unavailable if not collected at the time of their creation.
- *Dispensing:* providing access to their contents
- *Bibliographic:* identifying what exists on a topic
- *Symbolic:* conferring status and prestige on the institution

However, with the advent of *digital* and hybrid collections, these purposes are updated or at least extended with new dimensions. *Digital collection* is defined as a collection of virtual, digital and multimedia information resources. That is, a digital collection can be considered to contain multiple collections if it is able to access to remote digital libraries on behalf of its user community. As a result, such extension brings complexities

in collection management, such as "when libraries rely on cooperatively maintained digital libraries of metadata to determine what exists, where it exists, how to acquire access to it, and who is responsible for bibliographic control?" [42] (chapter 7). From this comment, obviously it can be seen that concerns are expanded to a 'group' level collection management, rather than viewing individual library collections as a sole actor in digital library applications. Digital artifacts may exist in a variety of digital collections in distinct formats so as to meet the requirements of their user communities, therefore cooperative agreements must be reached before acquiring digital objects on demand. In this thesis, one of our major tasks is to "collect" semantically related digital objects resided in multiple digital collections in diverse formats.

2.2.3 Metadata

There are multiple interpretations for metadata from simple ones, like "data about data", "any statement about an information resource", to analyzable ones, such as "the *value-added information* which documents the administrative, descriptive, preservation, technical and usage history and characteristics associated with resources" [43]. Because of the importance of metadata throughout the thesis, we study metadata from the aspects of purpose, role and usage respectively.

Purposes of Metadata

Due to various application scenarios, different library builders and users may have different objectives in applying metadata. In Figure 2.1, we illustrates some important 'purposes' for applying metadata [44, 45, 46, 47].

As shown in Figure 2.1, users can apply metadata to *find*, *identify*, *select*, *obtain* access, *reuse* and even *navigate* digital/physical entities. Some specific examples are: searching in a context for all documents on a given subject; identifying that an article sought by the user is from a prestige conference; selecting a collection of video clips that complement a specific textual document; and accessing a portal of copyright-protected journal articles. Moreover, metadata can be used to manage content and is essential to standardization. These functionalities enable recording information that will support future preservation activities and as well promote greater interoperability between heterogeneous metadata records.

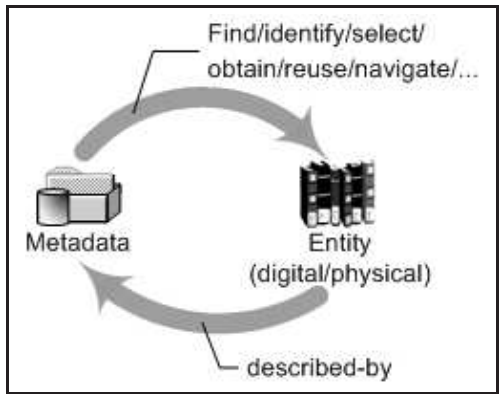

Figure 2.1: The Multiple Purposes of Using Metadata.

In this thesis, we focus mainly on using metadata to facilitate discovering semantically related resources instead of managing metadata records, and may ignore other issues to which it may legitimately be put, such as aforementioned preserving metadata in digital libraries. Thus, the purpose of applying metadata in our concern is more biased on facilitating and improving the retrieval of information. According to the definitions in the Information Retrieval (IR) theory, if too few relevant records are retrieved, we may have less recall; and if we are flooded by too many irrelevant results, we have poor precision. In fact, when users send queries to Google-alike search engines which largely adapt IR technologies, they *may* easily get very low precision (eg. "Tiger in Woods" vs "Tiger Woods"). The inverted index created by the search engines are generally statistical results which actually do not distinguish between documents having keywords as significant and incidental terms respectively. Metadata can help improve the precision in this concern by creating 'data' about the major content of the information resource, therefore it narrows down our searches onto collections which regard the keywords as important terms. For example, we could retrieve just those resources where "White" is the name of the author, without retrieving resources about "white house" or palette color. Metadata can also help enhance search recall by supporting retrieval of non-textual information in addition to textual documents. For example, images, audio, and PowerPoint slides can be retrieved if they are annotated by corresponding metadata.

Roles of Metadata

No single type of metadata can suit every such application, every type of digital object, and every community of users. Many literatures [7, 48] have discussed different roles of metadata and we summarize them as follows.

- *Content Independent Metadata:* This type of metadata does not describe any content information of documents, but it is still helpful for identifying documents. For example, 'publication-date' and 'saturation' used to capture the vividness of hue of a picture.

- *Content Dependent Metadata:* This type of metadata captures the content of the document it expresses. Examples are 'page range' of journal articles, 'rights' for accessing a document, and 'subject' in a play.

  - *Structural metadata:* it indicates how compound objects are put together, for example, how pages are ordered to form chapters.

  - *Administrative metadata:* it provides information to help manage a resource, such as when and how it was created, file type and other technical information, and who can access it.

  - *Descriptive metadata:* it describes a resource for purposes such as discovery and identification. It can include elements such as title, abstract, author, and keywords, etc. which can be divided into two sub-categories.

    * *Domain Independent Metadata:* They capture information presented in the document independent of the application domain. One example is the document type which can be HTML, XML, RDF or OWL files.

    * *Domain Specific Metadata:* They are highly associated with the specific content to the application domain. Thus controlled vocabulary, thesauri and ontologies[26] very important in this case as the terms have to chosen in a domain specific manner. Examples are 'relational databases and 'large text archives' in ACM Computing Classification System [49] which is from the computer science domain.

Use of Metadata

A classical example of using metadata is a library catalog card, which contains data about the contents and location of a book: It is data about the data in the book referred to by the card. Broadly speaking, use of metadata is to associate metadata with digital objects in digital libraries, such as indicating the 'source' or 'author' of described digital objects, and the 'type' and the 'rights' of how it should be accessed. To be more systematic, the methods of applying metadata can be divided into three categories [50]:

- Firstly, metadata information is directly *embedded* within the document itself. For instance, META tags in a HTML file are used to indicate metadata information and assigned in the HEAD part of Hypertext Markup Language (HTML) documents. From the theoretic perspective, such metadata information can be harvested automatically by web crawlers. Unfortunately, due to performance issue, most search engines which sending out the crawlers do not in practice extract META tags.

- As to the second method, metadata are maintained as an independent part which is *attached* to original resources. If one conducts a request, such a metadata record will be transferred back to him together with the resource. Intuitively, extra overhead must be considered in keeping a more complex data structure to accommodate metadata information and parser function to parse metadata during the transformation.

- Finally, metadata can also be stored physically apart from digital objects. Basically, library catalog cards can be categorized into this method. In current approaches, it becomes more and more common that metadata information maintained in a separated medium, such as an independent XML file or a third-party database. In our work, we consider only this type of metadata collection as well.

Due to the flexibility in associating metadata to resources, it easily leads to a hassle of applying meaningful metadata schemas. In a large scale application, often a significant number of communities may get involved. Similar or relevant digital objects can be annotated by different metadata schemas. In such a case, relationships between similar digital objects may never be exposed because there is not a mechanism to interrelate them.

Therefore, extra mechanisms, such as mapping, are needed to describe these relations which can be further processed to enable interoperation. We will come back to this issue in Chapter 5.

2.2.4 Identity

Libraries want to share content; publishers want to sell it. Museums strive to preserve culture, and artists to create it. Musicians compose and perform, but must license and collect. Users want access, regardless of where or how content is held [51]. One of the features that all of these stakeholders (and more) share is the need to have *identity*. The terms, such as "identifier" or "handler", are used to *identify* content and its owner, and to be able to share this information in a reliable ways that make it easier to find. Many different identifier schemes are available in both of the digital and physical libraries. For examples, the ISBN, a unique machine-readable identification number, has been used for 30 plus years and has 159 official members with the intention to mark any book unmistakably. As to identify periodical publications, an eight-digit number International Standard Serial Number (ISSN) [52] can then be used. In addition, this Serial Item and Contribution Identifier (SICI) standard defines a variable length code that will provide unique identification of serial items (e.g., issues) and the contributions (e.g., articles) contained in a serial [1] title [53].

Especially, in the networked digital library environment, identity is increasingly indispensable to enable persistent, reliable and location independent systems. Typical examples are: Digital Object Identifier (DOI) [54], Universal Unique IDentifier (UUID) [55], and URL/URN/URI [56] which stand for Uniform Resource Locator, Uniform Resource Names and Uniform Resource Identifiers respectively. The relations among "URI, URL, and URN" are clarified by the eventual URI Standard (RFC3986) [57] [58]:

[1] Serial may include periodicals, newspapers, annual works, reports, journals, proceedings, transactions and the like of societies and other corporate entities such as conferences, and numbered monographic series.

> *A URI can be further classified as a locator, a name, or both. The term "Uniform Resource Locator" (URL) refers to a subset of URIs that, in addition to identifying a resource, provides a means of locating the resource by describing its primary access mechanism (e.g., its network "location"). The term "Uniform Resource Name" (URN) has been used historically to refer to both URIs under the "urn" scheme [59], which are required to remain globally unique and persistent even when the resource ceases to exist or becomes unavailable, and to any other URI with the properties of a name.*

URI is first initiated by W3C and used as *pointers* to help users manipulating, imaging and finding their way in such a space. Some concrete examples are:

- mailto:Hao.Ding@idi.ntnu.no
- ftp://ftp.idi.ntnu.no
- http://www.idi.ntnu.no/index.php
- urn:oasis:names:specification:docbook:dtd:xml:4.1.2

The DOI is the outgrowth of a program for enabling *management of copyrighted materials* in an electronic environment [54]. Resolution of the DOI uses the Handle System [40], which offers the necessary functionality for open applications. As contrast to URIs which designate a location at which an instance is held, DOI allows the designation of instantial entities *directly*. A wider vision in DOI is to reduce the need for interoperable metadata by regarding each DOI instance as "just" a single-point resolution routing system, which holds any related information within a "proprietary" institution, such as a website. For example, two entities (i.e., items), a printed article and a PDF file, generated from a published article (i.e., manifestation) can be related by pre-defined identifiers.

A UUID is generally a 128 bit number assigned to any object which is guaranteed to be *unique*. The mechanism used to guarantee the uniqueness is through combinations of hardware addresses, time stamps and random seeds [55]. UUIDs require no central registration process and are fully compatible with the URN syntax.

Processing metadata information via unique identifiers saves our time in resorting to physical inspection of the items being requested. Most

digital library systems have used identifiers in annotating, storing, exchanging and reusing digital objects. It may alleviate metadata interoperability problem by holding relevant information together under one specific identifier, but a potentially infinite number of metadata associated with an identifier may be an impossible task. Moreover, identifiers will not consider how to inter-relate semantically relevant entities which do *not* belong to a common ancestral 'work' (i.e. according to the concepts in FRBR[46]).

2.3 Instances of Metadata Standards

2.3.1 MARC

MARC, which stands for MAchine Readable Cataloging, is the set of rules used by libraries for establishing a computer based catalog. Herein, "Machine Readable" means that one particular type of machine can read and interpret the data in the cataloging record, while "Cataloging" means that a bibliographic record traditionally shown on a catalog card. Led by Library of Congress[2], MARC is probably the most widely adopted communications standard for 'exchanging bibliographic, holdings, and other data' between libraries[60]. An example of MARC record is shown as follows: Generally, a MARC record includes: 1) a description of the item, 2) main entry and added entries, 3) subject headings, and 4) the classification or call number [61]. Often MARC records contain much additional information. In Figure 2.2, the 3-digit tags mark bibliographic *fields*, e.g., '100' and '245', which stand for personal name main entry and title information respectively. Most fields can be subdivided into *subfields* (normally lowercase letters) representing more detailed information[62]. For example, The MARC record in Figure 2.2 uses '260', '$a', '$b' and '$c' to indicate the 'publication area', with detailed information in 'place of publication', 'name of publisher', and 'date of publication'.

2.3.2 Dublin Core

In order to cope with a broad spectrum of applications that are emerging on the Web, a set of *simple* metadata standard is expected to be able to describe cross-domain information resources. The particular requirement on *simplicity* is due to the cost and infeasibility to use complicated metadata

[2]http://www.loc.gov/marc/

```
020 $a0-7456-3478-8
080uk$a655.41:004
082kj$a070.5797
082xd$a070.5
087ns$afa3827 082 070.5730941 083 Fa:10
100 $aThompson, John B.
245 $aBooks in the digital age
$bthe transformation of academic and higher education publishing...
$cJohn B. Thompson
260 $aCambridge
$bPolity Press
$c2005
300 $a468 s.
```

Figure 2.2: A MARC record (partial), generated from BIBSYS[1]

sets, e.g., MARC. Actually, there are 999 possible MARC tags, so it requires not only time, but also professional knowledge to develop metadata records. The Dublin Core metadata element set is invented under such background, with missions of cross-domain discovery, metadata interoperation and facilitating the development of community or discipline-specific metadata sets[63]. Actually, after 10 years development, the Dublin Core has become the *de facto* standard for cross-domain resource discovery metadata on the Web. The Dublin Core metadata elements(DCME) are summarized in Table 2.1.

Table 2.1 is depicted in two columns with elements(i.e. simplified dublin core) and corresponding refining elements (i.e. qualified dublin core). One exception is the element *audience* which is not in the original 15 elements, but has a recommended refining element *mediator*.

The success of the Dublin Core metadata set is greatly due to its simplicity and easy adaptability. More precisely, as remarked in [64], "it is a small language for making a particular class of statements about resources. Like natural languages, it has a vocabulary of word-like terms, the two classes of which – elements and qualifiers – function within statements like nouns and adjectives; and it has a syntax for arranging elements and qualifiers into statements according to a simple pattern".

Table 2.1: Summary of Dublin Core. Further details from [9]

DCME element	Element Refinements (Qualifiers)
audience	mediator
contributor	-
coverage	spatial, temporal
creator	-
date	alternative, available, created, issued, modified, valid
description	abstract, tableOfContents
format	extent
identifier	-
language	-
publisher	-
relation	isFormatOf, hasFormat, isPartOf, hasPart, isReferencedBy, references, isReplacedBy, replaces, isRequiredBy, requires, isVersionOf, hasVersion, confirmsTo medium
rights	-
source	-
subject	-
title	alternative
type	-

2.4 Search Protocols in Digital Libraries

As digital libraries pervade the online system, great needs are shown in interacting various different content of digital libraries. Basically, the interaction can be implemented in different mechanisms: 1) Sockets, which are typically supported by low-level Internet operations; 2) HTTP, an application-level protocol for distributed, collaborative and hypermedia information systems on the Web; 3) CORBA IIOP, a high-level architecture that supports a distributed object-oriented paradigms; and 4) SOAP, a XML-based lightweight protocol, which can be further combined with a variety of other protocols, e.g. HTTP.

Two prominent protocols used in digital libraries are the Z39.50 [65] protocol and the Open Archives Initiative (OAI) protocol [66]. There are also other important protocols developed in research projects, such as Dienst protocol [67] and Stanfords Simple Digital Library Interoperability Protocol (SDLIP) [68, 69].

OAI and Z39.50 protocols can be regarded as two ends of a spectrum. Z39.50 has been widely used in large digital libraries. But small- or moderate-sized institutions seldom apply Z39.50 due to the costs of complexity. OAI is engaged in determining a proper degree of modesty, which balanced the need for adequate functionality against the requirement that the cost of entry for participating archives be sufficiently low [70].

2.4.1 Z39.50

The Z39.50[65] defines a wide-ranging protocol for *client-server* based information retrieval. It specifies procedures that allow a client to search a database provided by a server, retrieve database records, and perform related information retrieval functions.

The protocol does not address interaction between the client and the end-user. Instead, it defines the protocol control information, the rules for exchanging this information, and the conformance requirements to be met by implementation of this protocol. To address the broad issue of accessing and retrieving heterogeneous data in different domains, Z39.50 includes a set of classes, called *registries*, which provide each domain with predefined structure and attributes. Registries cover query syntax, attribute fields, content retrieval formats, and diagnostic messages. For instance, various MARC fields are specified in the content retrieval formats.

The Z39.50 protocol is divided into 11 logical sections - *facilities* that each provides a broad set of services. Table 2.2 describes a summary of each facility.

Table 2.2: Facilities in Z39.50. From [10] pp.428

Z39.50 Facility	Client-side description
Initialization	Establish connection with server and set/request resource limits.
Search	Initiate search using registered query syntax, generating a result set on server-side.
Retrieval	Retrieve a set of records from a specified result set.
Result-set-delete	Request deletion of server-side result set or sets.
Access control	Server initiated authentication check.
Accounting& Resource Control	Request status reports of committed server resources and dictate if server is allowed to contact client when agreed limits are reached.
Sort	Specify how a result set should be sorted.
Browse	Access ordered lists such as title and subject metadata.
Explain	Interrogate server to discover supported services, registries, and so on.
Extended services	Access services that continue beyond the life of this client-server exchange, such as persistent queries and database update.
Termination	Abruptly end client-server session.

As implied in Table 2.2, although it is not necessary to implement all parts of the protocol, it is indeed a daunting work to conduct a full implementation and may be inappropriate for a moderate-sized digital library. To deal with such a problem, a minimal implementation is specified in terms of the initialization facility, the search facility, the retrieval facility (partial) and the explain registry (i.e. registry).

Recently, the Search/Retrieve Web service (SRW) protocol, based on the Z39.50 protocol, has been released [71]. The primary goal of the SRW

protocol is to support interoperable Web service that handles information retrieval in distributed networks. It uses client/server architecture as well.

2.4.2 The Open Archive Initiative

Another interesting approach is the Open Archives Initiative (OAI) [66], a technical framework not intended to replace other approaches but to provide a low barrier, easy-to-implement and easy-to-deploy alternative for different participants than other complex protocols, such as Z30.50 or specific ones, such as Google API (c.f. http://www.google.com/apis/).

The protocol supports interaction between a *data provider* and a *service provider* which is in essence the client/server model. However, it emphasizes in client-driven interaction. That is, the client decides what services are offered to users. In contrast, data providers focus on managing repositories. They do not have to process user queries concerning searching over specific records, instead, they must support those that have content with fixed metadata records, those that computationally derive metadata in various formats from some intermediate form or from the content itself, or those that are metadata stores or metadata intermediaries for external content providers [72]. Of course a digital library may choose to be both a data provider and a service provider. However, it has to maintain more than one repository.

The general facilities supported by the OAI protocol are embodied in six verb arguments. Table 2.3 summarizes them.

Table 2.3: OAI Protocol Requests

OAI protocol request	Description
GetRecord	Returns the repository item specified by the document identifier in the requested format.
Identify	Return both fixed format and domain specific descriptions.
ListIdentifiers	Return a list of document identifiers.
ListMetadataFormats	Return the metadata formats supported by the repository in general or for a specific document.
ListRecords	Returns a list of repository items in the requested format.
ListSets	Returns the repository's classification hierarchy.

GetRecord is used to retrieve an individual metadata record from a repository. *Identify* and *ListSets* are typically called in a client's interchange with a server to have a general information of the repository. *ListIdentifiers* is a way of receiving all the document identifiers or a group that matches a specified set name. It retrieves only headers rather than

records. *ListMetadataFormats* is used to retrieve metadata formats available from a repository as a whole or to a particular document within it. Unqualified Dublin Core is mandatory, but other formats such as MODS and MARC, which are able to describe metadata per record in a greater granularity, may also be supported. *ListRecords* is used to harvest records from a repository - as different from *GetRecord*, more than one record can be returned.

Great flexibility can be achieved using the verb arguments. Service providers can retrieve in a distributed and dissimilar environment records in different granularities. Note that *semantics* in metadata are *not* considered in the protocol and are maintained by individual libraries/archives [70]. Moreover, OAI is built upon client/server architecture as well.

2.4.3 Dienst

The Dienst protocol[3] provides for communications with services in a distributed digital library. It has three facets: 1)A conceptual architecture for distributed digital libraries; 2)A protocol for service communication in that architecture; 3)A software system that implements that protocol [67].

This protocol supports search and retrieval of documents, browsing documents, adding new documents, and registering users. Each of these is an independent service and a digital library collection can simply combine these services. There are six categories of services described as follows:

- A *Repository Service* stores digital documents and associated metadata.
- An *Index Service* accepts queries and returns lists of documents identifiers matching those queries.
- A *Query Mediator Service* dispatches queries to appropriate index servers.
- An *Info Service* returns information about the state of a server hosting one or more services.
- A *Collection Service* provides information on how a set of services interact to form a logical collection.

[3] http://www.cs.cornell.edu/cdlrg/dienst/DienstOverview.htm

- A *Registry Service* stores information about (human) users of services of a collection.

Communication with and among individual Dienst services takes place via an open protocol, which makes it possible to combine the services in innovative ways, or build other service layers on top of the basic Dienst services. In fact the Dienst was a source of inspiration for the OAI designers.

2.4.4 Simple Digital Library Interoperability Protocol (SDLIP)

Simple Digital Library Interoperability Protocol (SDLIP) [73], at the Stanford University, is derived from the original CORBA-based Digital Library Interoperability Protocol (DOLIP). There are two levels of SDLIP capabilities: SDLIP-Core and SDLIP-Asynch. SDLIP-Core implements synchronous operations only and clients invoke search operations on servers, and 'hang' until the operations return with the result. The second level, SDLIP-Asynch adds the ability for clients to invoke search operations that return immediately. Services then deliver result information back to the client through one or more callbacks. SDLIP-Asynch thus subsumes SDLIP-Core.

The main goals of SDLIP are as follows [73]:

- Simplicity for both client and server side implementations;
- Implementations possible via both distributed object technology, such as CORBA, and via HTTP;
- Support for stateful and stateless operation at the server side;
- Support for dynamic load balancing in server implementations;
- Support for thin clients, such as handheld devices.

To use SDLIP protocol, servers need not implement all these options, such as synchronous and asynchronous interactions between client and server. It is up to a client to establish what functionality is supported by applying the protocol. SDLIP places stress on a design that is scalable, permitting the development of digital library applications that run on handheld devices, such as portable digital assistants (PDAs) , as well as workstation and even mainframe-based systems. The two transport options, CORBA and HTTP, can be used respectively or mixed freely.

Herein, we emphasize here that SDLIP protocol is also client/server based, although every service like search, attribute translation or metadata service is a distributed object implementing a well known interface (i.e. CORBA).

2.5 Information Searching in Digital Libraries

Half a century ago, librarians had to work hard manually on *card catalogue* or standardized classification schemes, such as the Dewey Decimal Classification[74], in order to support bibliographic search. With the advent of Web, various approaches have thus been initiated and extended for information searching in digital libraries. We introduce three typical approaches used in digital libraries, namely, traditional keyword-based IR, metadata-based search and ontology-based search. The last two approaches are complementary to the first one, rather than separated techniques.

2.5.1 Keyword-based Information Retrieval

Keyword-based information retrieval is often regarded as synonym of *document retrieval* and nowadays, with *text retrieval*, implying that the task of an IR system is to retrieve documents or texts with information *content* that is *relevant* to user's queries [75]. The Vector Space Model (VSM) [2], as a standard technique in keyword-based search, represents documents through the words they contain and helps decide which documents are similar to each other and to keyword queries. The VSM is probably the most widely adopted *non-semantic* technique, while other techniques, such as stopwording, word stemming, text mining, have been conducted to reduce the effects caused by term usages and add 'intelligence' to entire systems. The exact *meaning* of terms have not been considered in the entire IR procedure. For example, if there is a query - 'Henry Ibsen', the system might return a document containing '*Henry* George Smith' and 'Zak *Ibsen*'. Generally, keyword-based IR involves two related but different processes: *indexing* and *searching* (c.f. Figure 2.3).

Basically, indexing denotes processing and expressing documents or user queries for retrieval purposes. For example, documents will be parsed and represented in a set of ordered items. Searching refers to looking in the parsed document items for the occurrences of all words and patterns

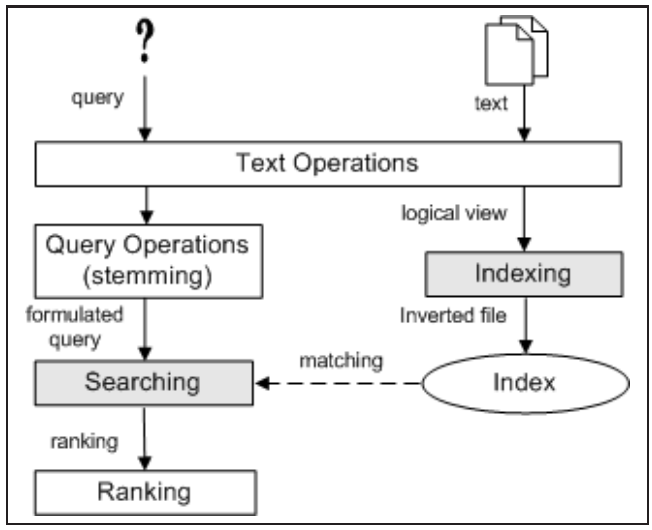

Figure 2.3: The Generic Retrieval Process. Adapted from [2] (pp.10)

present in the query. In practice, how the indexing and searching processes can be carried out may be different in some particular way.

2.5.2 Metadata-based Search

As an extension to the VSM which has to process entire document to realize comparison to terms in user queries, metadata-based search focuses on specific metadata fields instead of the entire documents. Basically, those metadata records are the mental understandings on corresponding documents, and often involve a process of tagging the documents with meta-level signs (i.e. metadata) in an explicit way. Figure 2.4 illustrates a simple metadata creation and search process.

The forms of metadata searches are not static. From the perspective of targeted users, it can be categorized as *comprehensive search*, *known-item search* or *searches for facts* [15]. The comprehensive search is mainly applied in scientific or scholarly domains where users want to discover relevant work in a specific field. Users applying the known-item search are usually looking for specific records containing words in specified metadata fields, such as the query "author = 'Henry Ibsen' returns and only returns all plays written by 'Henry Ibsen'". The "facts" in *searches for facts* are "specific items of information that may be found in many sources of

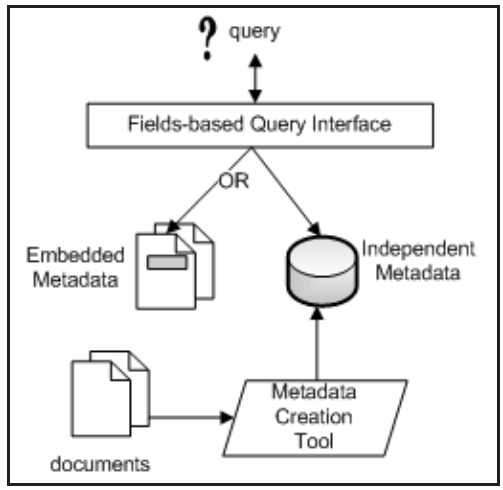

Figure 2.4: A Simple Metadata Creation and Search Process.

information" [15]. That is, a search for a fact may cover many possible sources. Thus, if relevant resources have explicit relations exposed (i.e. interconnected) and provided that one useful item has been found, then other related resources may be retrieved as well.

However, few approaches support such "intelligent" queries by far. The reasons may be partially due to inaccurate data in a field, misspelling and typos, whereas it is largely because of inadequate metadata descriptions and heterogeneous metadata schemas. Few approaches can afford complete metadata descriptions because of considerable human effort, experience, and cost to index and describe resources. Therefore, many related resources may not be found even if they are implicitly "interconnected".

2.5.3 Ontology-based Search

Different kinds of applications require different levels of details in metadata. For conventional IR, a simple string search can often find a document with the desired information. To find information about *plays written by Ibsen*, it could search for the strings 'Ibsen' and 'Play'. Conventional IR system depends on a human reader to decide which strings to search for and to interpret the results that are retrieved, while requiring also 'intelligence' to distinguish *Ibsen's Plays* from a *game* that a person named Ibsen takes part in.

Ontology-based search, from the semantics' perspective, provides added values in searching over documents which are semantically related, while not just a matchmaking service which may lead to wrong results. More detailed discussion on Ontology is in Section 6.3.2. Herein, as remarked by McGuinness[76], the usages of ontologies in information search are as follows:

- *interoperability*: we may have a complete operational definition for how one term relates to another term and thus, we can use equality axioms or mappings to express one term precisely in terms of another and thereby support more "intelligent interoperability";

- *structured, comparative, and customized search*: if an ontology contains mark-up information it can be used to prune comparative searches and to point which properties are most useful to present in comparative analyses so that users may have concise descriptions of the resources instead of comparisons in complete detail;

- *generalization/specialization*: If a search application finds that a user's query generates too many answers, one may dissect the query to see if any terms in it appear in an ontology, and if so, then the search application may suggest specializing that term.

In ontology-based search, ontologies are used to infer implicitly hidden information between concepts Hence, user queries may be reformulated and resources may also be re-processed (e.g., semantic annotation) such that more formally defined semantic information can be appended to the original resources. A simple ontology processing component is illustrated in Figure 2.5.

In addition, recent approaches [77, 78] also introduce rule inferencing into the ontology-based search due to the description limitations in current ontology languages. As a plus, domain specific rules defined by domain experts (manually or by tools) can infer more complex high-level semantic descriptions, for example, by combining low-level features in local repositories. On one hand, the rules can be used to facilitate the task of resource annotation by deriving additional metadata from existing ones [77]. On the other hand, during the process of query reformulation, the rules can be used to help define constraints on queries. Obviously, both approaches make an important step towards retrieving higher quality records. To be coherent, we extend Figure 2.5 into Figure 2.6 shown as follows:

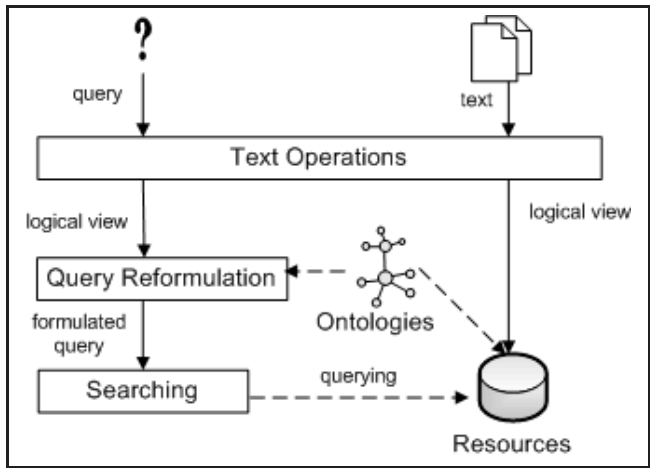

Figure 2.5: A Simple Ontology-based Search Process.

2.5.4 Federated Search Solutions

Federated searching is an enhanced mechanism for searching over a range of different metadata formats by combining several search protocols, such as Z39.50, XQuery and OAI. Baeza-Yates & Ribeiro-Neto (1999) give their definition of the concept[2](page.442).

> "**Federated Search:** finding items that are scattered among a distributed collection of information sources or services, typically involving sending queries to a number of servers and then merging the results to present in an integrated, consistent, coordinated format."

The requirement for federated searches originates from the 'holy grail' in digital library work, which aims at seamless federation across multiple distributed digital libraries, providing 'one-stop' library services for their patrons. However, greatly distributed and heterogeneous resources make effective search and discovery problematic and challenging. An exemplar of federated searching is shown as below:

As illustrated in Figure 2.7, a system supporting federated searches has to be able to accommodate various distributed information resources, such as full-text repositories maintained by commercial and professional society publishers; preprint servers and Open Archive Initiative (OAI) provider sites; specialized Abstracting and Indexing (A&I) services; pub-

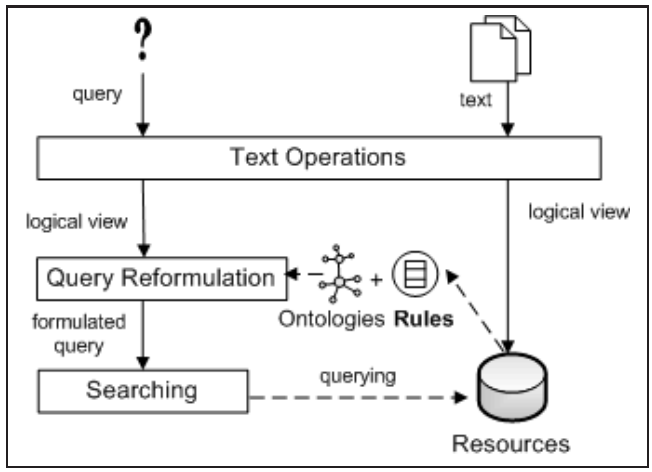

Figure 2.6: An Ontology-based Search Process Powered by Rule Inferencing.

lisher and vendor vertical portals; local, regional, and national online catalogs; Web search and metasearch engines; local e-resource registries and digital content databases; campus institutional repository systems; and learning management systems. Generally, the major features of federated searches, which is determined by the characteristic of *federation*, can be summarized as below:

- searches have to be deployed across disparate information stores;
- responses from these disparate systems have to be collected and collated;
- results are generated in a *single* list back to the user.

In order to realize such features, a custom made search protocol is required for a closed system consisting of homogeneous search servers and particularly if users require special functionality such as encryption of requests and results. Otherwise, a standard search protocol may be used for the benefit of a more easily interoperation with other search servers for the system[2].

Weaknesses in this approach come in two-fold. First, as illustrated in Figure 2.7 (cf. middle layer in grey rectangle), system can not scale well since increased user queries and participated digital library systems may

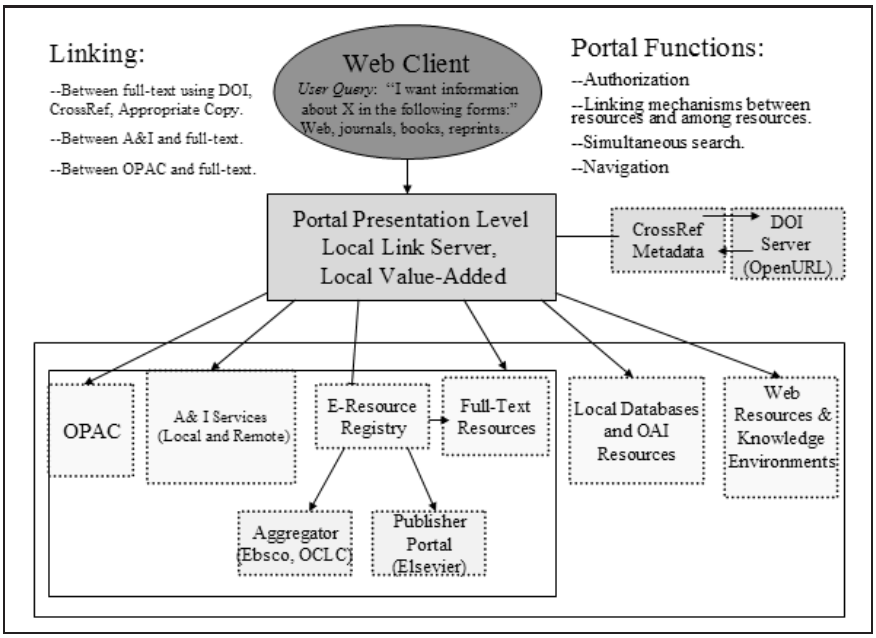

Figure 2.7: Federated Search Diagram, from [3]

easily deplete limited resources on central servers. One practical example can be found in the early usage of NCSTRL digital library [15]. Second, one can not get better results with a system supporting federated search than with an average database system. All federated search does is to interpret translate a search into something the native digital library can understand. Not surprisingly, limited capabilities of some native digital libraries may constrain the fulfillment of such query transformation. For example, a federated search cannot do a three-term search with Boolean operators in a native database whose interface doesn't support it. In a word, federated searching cannot improve on digital libraries' search capabilities - it can only use them[79]. Other challenging tasks in federated searching, although out of the scope of this thesis, have been recognized in ranking the documents received from several distributed sources and de-duplicating redundant copies stored in different digital libraries [80].

Approaches have been conducted for dealing with the weaknesses in federated searching, such as using an integrated *representative* for databases to indicate their contents[81]; and creating *concept spaces* to achieve semantic indexing and searching over specific domains (ie. engineering,

physics and computer science)[82]. However, these approaches are in practice too expensive for small and moderate digital libraries. For example, DELIVER digital library at UIUC has used supercomputers to compute concept spaces for progressively large collections[82]. Currently, researchers are also trying to introduce new system infrastructure (eg Peer-to-Peer networks) and standardized description languages (eg. XML, RDF and OWL) to achieve both system scalability and search precision. Focused on this direction, we will in the rest of this thesis investigate a semantic search framework in Peer-to-Peer based digital libraries.

2.6 Challenges in Distributed Information Search in Digital Libraries

Effective search and discovery over open and hidden digital resources on the Internet scale remains a problematic and challenging task [3]. In large-scale federated digital libraries, keyword-based search prevails in current approaches while less efforts are put in supporting semantic search. Although some advanced approaches [82, 83, 84] have been conducted by applying ontologies to specify and process subsumption relations between relevant metadata elements, few of them has based their approaches on large scale distributed systems, especially in situations where many heterogeneous metadata schemas are applied, such as that in Semantic Web.

We recognize that there is much research to be done in this area and thus focus only on following two major perspectives. From the perspective of the digital library architecture, current typical client-server and 3-tier architectures show incapacity to meet specific requirements in system scalability, flexibility, and inter-communications (if without central server's control). It is greatly due to largely distributed and *independent* information resources. The Z39.50, OAI, Dienst and SDLIP protocols are all client/server oriented approaches and suffer the scalability and single-point-of-failure problems aforementioned. The first version of Dienst protocol used in NCSTRL has suffered system crash because of overloading on servers[15](pp.218). Thus, special efforts are needed in exploring novel architectures [47] which are beyond current base of deployed protocols and system infrastructures. Among these approaches, systems built upon Peer-to-Peer overlay networks hold many promises in alleviating the problems in terms of system infrastructure [85, 86, 87]. At the same time, Peer-to-Peer based applications also raise some complex ques-

tions, such as how to deploy prototype systems and the tradeoffs between advanced capabilities and ubiquity of access. More discussion on this issue is in Chapter 3 and Chapter 4.

On the other hand, difficulties exist in coping with metadata heterogeneities, namely, system heterogeneity, syntactic heterogeneity, structural heterogeneity and semantic heterogeneity [88]. An ideal approach would be to develop a comprehensive set of standards that all digital libraries would adopt. However, this notion fails to recognize the costs in adopting these standards, especially in times of rapid change. In this thesis, we focus intensively on semantic heterogeneity which is regarded as the most fundamental and most complex issue in interoperation. As Madnick has noted, large-scale semantic heterogeneity is caused by that "each source of information and potential receiver of that information may operate with a different context"[89]. Therefore, in order to achieve "mutual understanding" in the context among participants, specific efforts are required in explicating relations between relevant terms in different metadata sets. For example, creating "semantic bridge" to map descriptions between similar terms [24, 90, 91], translating one term to another by referring to commonly shared/understood vocabularies [92, 93, 94], or simply generating a globally shared ontology [95]. However, none of these approaches turn out to be a trivial problem and we are to extend the discussion in Chapter 5, Chapter 7 and Chapter 8.

2.7 Chapter Achievement

- Introduced the definition of digital libraries and basic building blocks of digital libraries, namely, collection, metadata and Identity.
- Described searching protocols used in digital library systems: Z39.50, OAI-PMH, Dienst and SDLIP.
- Described information searching strategies in digital library systems: keyword-based search, metadata search and ontology based search. Particularly, federated search has been presented and justified as not suitable for semantic search in large-scale distributed and heterogeneous digital libraries.
- Discussed the challenges in searching heterogeneous records in distributed digital libraries from two perspectives, namely digital library architecture and semantics.

Chapter 3

P2P Overlay Network and Digital Libraries

3.1 Introduction

This section is devoted to investigating possible solutions for latent problems in current digital library architecture. In this work, we concentrate on the Peer-to-Peer overlay networks which are directly involved in the physical communication among digital library systems. We are to introduce typical P2P models and describe briefly the functionalities of several practical P2P systems which receive enormous concerns in building large-scale distributed systems.

3.2 Why P2P in Digital Libraries?

3.2.1 Digital Library Architecture of the Past and Current

All digital libraries share a certain general characteristic of assembling a collection of materials needed by their users. Due to the huge varieties in coverages, subjects, representations and operabilities (e.g. operation systems), a single library may not provide all what users want. Naturally, many mutually interested digital libraries would need to find a way to cooperate such that their collections can be shared in a broader horizon. From a theoretical perspective, to integrate disparate systems into a holistic one could solve the problem, but from a technical perspective, it is almost infeasible to conduct such integration because of the *specialization*

aforementioned.

In the 1990s, when client/server computing architecture became at the peak of its popularity, some projects integrate digital libraries in an ad-hoc mechanism (c.f. Figure 3.1) to realize a *loosely coupled* federation among them.

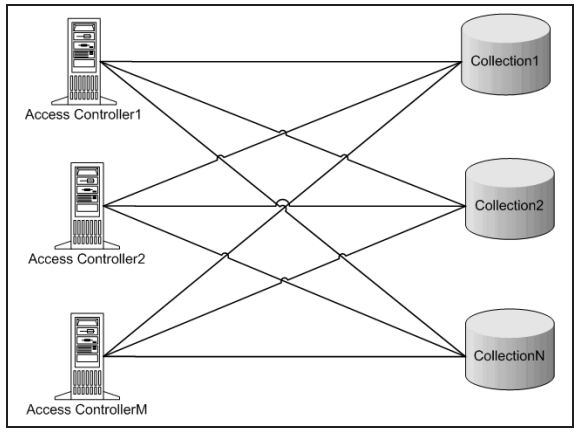

Figure 3.1: The Ad-hoc Digital Library Architecture

In such an ad-hoc model, *access controller* (or access services) are required to execute a standard protocol so as to reach disparate collections. One practical example of this model is the Networked Computer Science Technical Reference Library (NCSTRL) [96] where Dienst protocol [67] (c.f. Section 2.4.3) was used for communication with distributed digital library servers, allowing a user to access complete documents, metadata or named sub-parts. Often multiple interfaces have to be maintained for these collections which may make the distributed application very fragile [97]. In fact, although administrative services, e.g. naming services, can simplify integration by reducing and standardizing the interfaces between components, it is still difficult for themselves to achieve maximum adaptability and flexibility. Furthermore, another important weakness of this model, as reported in [15], is that the system can not scale well with a large number of servers.

To provide a uniform interface to distributed collections of the digital libraries, a layer of 'broker' (c.f. Figure 3.2) is created to integrate together the storage, delivery, searching, and browsing of distributed collections. By wrapping the digital library's core services inside a middleware layer, existing and new resources can be more easily integrated into the digital

library. In this type of system, scalability depends on the middleware architecture that supports communication between components.

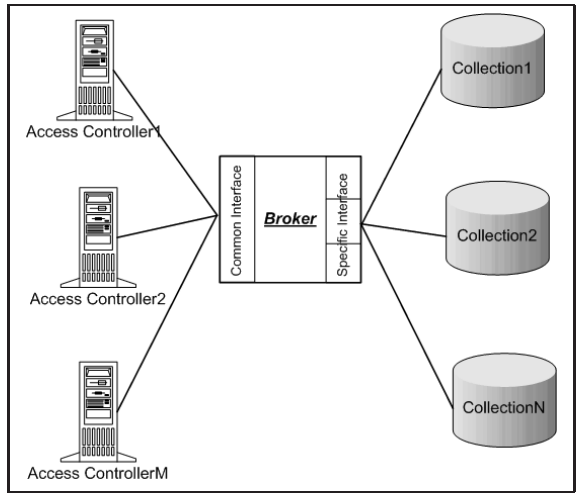

Figure 3.2: A Middleware-based Digital Library Architecture

The Simple Digital Library Interoperability Protocol (SDLIP) [73] (c.f Section 2.4.4) can be regarded as a 'broker' model where clients use SDLIP to request searches to be performed over information sources.

These digital library architectures, mainly derived from the client/server model, broke down the regime of monopoly of a few data providers across the world and encouraged resource sharing. However, there are still some critical weaknesses concerning the architecture issue:

- Performance bottleneck: servers become a bottleneck when too many requests are sent across the network.
- Dependence: all clients depend heavily on servers. That is, clients have to trade their *autonomy* to reach an agreement with each other, such as accessibility and data structure.

To alleviate such weaknesses, one basic requirement is to retain a scalable, adaptive and configurable infrastructure. These digital libraries shall also keep a considerable high degree of autonomy. Hence, innovative mechanisms are required to support loosely coupled digital libraries. To this end, any approaches in aspects of architecture, workflow or functions are sound if they can seamlessly 'integrate' activities and modules into a coherent whole, such as a generic and modular digital library architecture.

3.2.2 Digital Library Architecture of the Future

The architecture of the future digital libraries, as DELOS outlined in [85], should be able to allow any users to access available information/knowledge resources from anywhere and at any time and in an effective and efficient manner.

Among current research activities and developments, three kinds of architectural[1] approaches receive more attentions, namely: P2P overlay network, Grid computing [98], and Service-Oriented Architecture (SOA) [99, 100]. [85] has summarized the utilities of introducing these approaches into digital libraries as follows:

- Peer-to-Peer Architectures: It allows digital libraries to share collaborative data among them and a loosely coupled integration. Different aspects of peer-to-peer systems (e.g. indexes, and P2P application platforms) have to be combined and integrated into an infrastructure for digital libraries.

- Grid Architectures: As initially invented for intensive computing by 'clustering' distributed computing and data resources into a virtually single system, Grid computing architectures can help process certain services within digital libraries which are complex and computationally intensive (e.g., calculation of certain features of multimedia documents to support content-based similarity search). In addition, it also retains the flexibility to work on multiple smaller problems.

- Service-oriented Architectures (SOA): SOA is a paradigm for designing, developing, deploying and managing discrete units of logic (services) within a computing environment. Various functions provided by certain libraries, such as searching, navigation, and preservation, can be wrapped as Web services which provide us a standard and alternative way to share data and documents. In addition, existing services can be reused or composed for further more complex applications.

Among these approaches, Grid Architecture aims to solve problems too big for any single computer (eg. a digital library system), whilst

[1]Here we refer to a broader meaning of 'architecture' which may indicate both of the physical and logical based architecture.

retaining the flexibility to work on multiple smaller problems; SOA requires developers to think beyond the boundaries of their application and consider reusing existing services. These two approaches are out of the scope of this thesis which focuses on applying appropriate network topologies/infrastructures in constructing digital library systems. Thus in this thesis, instead of extending the scope wider to grid computing and SOA technologies, we focus on studying various P2P models and justify how and to what extent they can applied in the world of digital libraries.

As against the client/server architecture, the greatest strengths of P2P overlay networks are their decreased dependency on the server and their decentralization of control from servers. In addition, some P2P models do not require servers or put them to a bare minimum. A direct intuition is that to share collections among digital libraries, users would not have to seek the help of the server, as they can do this directly among themselves. However, as elicited in the first research question, efforts are required to investigate the feasibility of applying P2P architectures in building digital libraries. In the rest of this chapter, we investigate various P2P models and typical P2P systems in current research. Further approaches are to be conducted in Chapter 4.

3.3 Various P2P Models

3.3.1 Pure P2P Model

The pure P2P-based model breaks the conventional method of communication in client/server-based models in which the entire communication process between the client and server takes place based on *rules* the server sets. This model depends entirely on computers (i.e. clients in the client/server model) and works without relaying on any central server (see Figure 3.3). Herein, we use the term **pure** to suggest servers not involved in networking. Once such P2P application is running in the machine, peers find other connected peers on the network dynamically. The subsequential communications, such as, sending requests, receiving responses, uploading and downloading files, and conducting online activities, occur among connected peers without any assistance from a server.

Pure P2P models provide almost plug-and-play features for working with the Internet, in the sense that one can just connect to the Internet and she can use the P2P feature. Another advantage of the pure P2P model is that it is beneficial for not only the Internet but also an intranet.

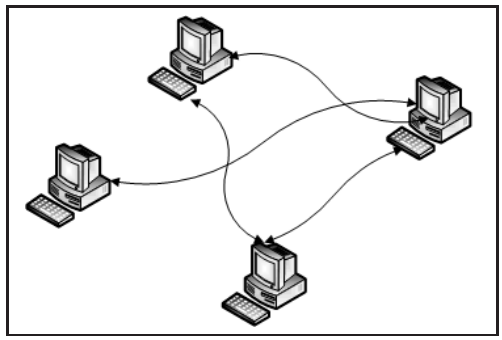

Figure 3.3: The Pure Peer-to-Peer Model.

The only problem with the pure P2P model is finding peers on the network. Because no central administration registers the entry of peers that log in to the network, the users themselves have to locate other peers. Fortunately, many approaches have been conducted for locating peers efficiently. In essence, the pure P2P-based model allows domain experts to set up reasoning *rules* and set up their own networking environments, so the problem is evolved to designing appropriate and efficient *rules* for peer discovery. For examples, a simple query flooding system architecture can be applied, such as Gnutella [101]. Moreover, if higher efficiency is to be achieved, other approaches, such as the distributed hash table (DHT) [102], Routing Indices (RI) [103] and Semantic Overlay Network (SON) [104, 105], can be applied. Note that the system behaviors or the degree of dependence among peers may vary by applying specific 'rules'.

3.3.2 Hybrid P2P Models

P2P with Simple Discovery Server

Such P2P models do not actually include a server. Although 'server' is used in this model, the role of the server in this model is restricted to providing a list of already connected peers to the incoming peer. Note that the connection establishments and communications remain to be P2P (c.f. Figure 3.4).

The major advantage of such P2P model over the pure P2P model is the increased chances of finding a larger number of peers on the network. To download a resource, a peer has to approach each connected peer individually and post its request, which in turn makes the process

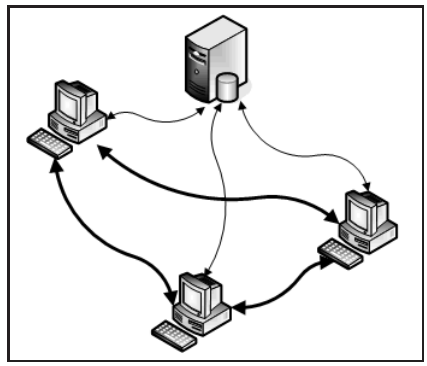

Figure 3.4: The P2P with Simple Discovery Server Model. Only the discovery of clients occurs via the server; the rest of the communication occurs among peers

time consuming. In contrast, in the client/server-based models, any peer looking for resources does not need to visit other peers, as the server itself maintains all the required content.

P2P with a Discovery and Lookup Server

In this model, the server is used to provide the list of connected peers along with the resources available with each of them. Hence, this model integrates the features of the pure P2P and the P2P with simple discovery server models for enhanced functionality of the server.

This model reduces the burden on peers, as there is no longer a need to visit each peer personally for the required information. The server in such a model initiates communication between two peers; again, the two connected peers establish communication, keep it alive, and perform various activities, like logging into the database in the connected peers, entering an index of resources shared by them, and so on.

P2P with a Discovery, Lookup, and Content Server

In this model, the server dominates as in typical client/server architecture. All the facets of catering to the requests of peers are moved from the peers to the server (c.f. Figure 3.5).

As shown in Figure 3.5, peers are not permitted to connect with each other directly, as all resources are stored in the centrally located server. If a

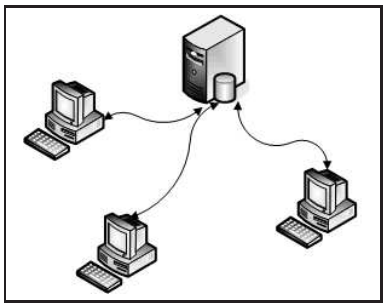

Figure 3.5: P2P with a Discovery, Lookup, and Content Server

peer requires information, instead of communicating with another peer, it approaches the server. The server processes requests and returns answers to the peers. The major disadvantage of this model, as similar to the client/server model, is that the server slows down if too many requests are generated simultaneously. Another disadvantage of such models is high cost because the server has to manage and store data and process all requests by itself. Because such models are entirely dependent on the central server, chances of single point failure increase. This is basically not the case with the previous P2P models.

3.3.3 Super-Peer based P2P Model

The very name of this model suggests its content. A super-peer P2P model, from certain perspective, can be regarded as an integration of the pure P2P model and the client/server model, which in turn forms a two-tier P2P model (c.f. Figure 3.6).

As illustrated in Figure 3.6, the particular nodes given the super-peer tags operate as both server and client to a set of clients and an equal in a set of super peers. Within the purview of a set of clients, super peers act as servers providing services, such as listing connected peers, acting as primary connection nodes and sometimes search hubs. At the higher level, super peers actually form a pure P2P model, where different connection/communication policies can be applied. Practical systems in such models are JXTA [106] and Kazaa [107], both of them provide the efficiency of centralized network as well as autonomy, reliability and load balancing of distributed network.

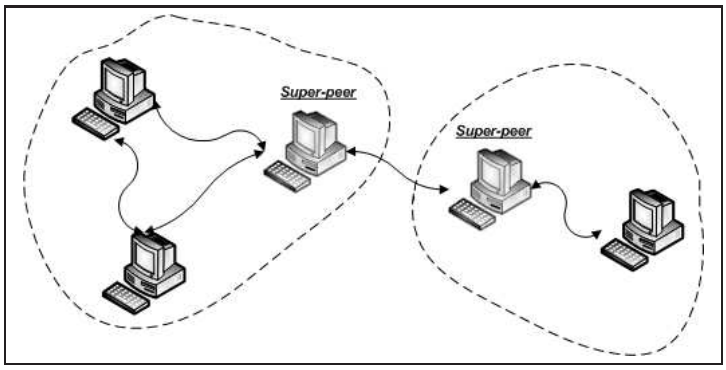

Figure 3.6: Super-Peer based P2P Model

3.4 Existing P2P Systems

3.4.1 Gnutella, Naspter and Freenet

Gnutella [101], Napster [108] and Freenet [109] are ancestors in P2P computing. They all support and only support keyword-based search. Gnutella is a representative instance for query flooding which does not scale well. Napster goes in the opposite direction and adopts central servers to maintain a centralized directory from which connected peers can register their profiles/expertise and also retrieve a list of peers matching user's request. In Freenet, each file/document is identified by a binary key which is generated using hash function; each peer maintains a local routing table which keeps information about neighboring peers and the keys are a sequence of *(file key, node address)* pairs used for retrieval. Lookups in Freenet take the form of searches for cached copies. This allows Freenet to provide a degree of anonymity, but prevents it from guaranteeing retrieval of existing documents or providing low bounds on retrieval costs.

3.4.2 Routing Indices

Crespo [103] uses Routing Indices (RIs), created and maintained by each peer, to forward queries to neighbors that are more likely to have answers. If a neighboring peer cannot answer a query, it forwards the query to a subset of its neighbors by referring to its local RI, rather than by selecting neighbors at random or by flooding the network by forwarding the query

to all neighbors. Since RIs are created and maintained individually, any updates on RI, such as peer joining or leaving, may lead to a cascade of updates in peer network. This is also the overhead RIs generated for the sake of efficient query forwarding in RIs instead of random flooding. RI-based systems can also specify relations between query 'topics' and neighboring peers, which moves further than pure keyword-based query.

3.4.3 Distributed Hash Table (DHT)

Distributed Hash Table (DHT) is probably the most widely used algorithm in P2P computing. DHT specifies a relation between entities (eg. files) and an identifier (ID) in P2P network. That is, DHT systems assign each entity (e.g. file names) a *key* generated by a hashing algorithm, then map the key to the node which also has an *ID* (e.g. hashed IP address). Normally this ID is the one closest to the key, and the storage and lookups of keys are distributed among multiple hosts. Normally, communication cost and the state maintained by each node scale logarithmically with the number of nodes N. That is, each node maintains information only about $O(logN)$ other nodes, and thus a lookup will require $O(logN)$ messages. DHT is probably the most efficient and applicable algorithm so far and performance of all DHT algorithms has been reported pretty good [110].

Representative DHT systems are Chord [111], CAN [112] and pSearch [113] . It is reported that these systems can adapt efficiently as nodes join and leave the system, and can answer queries even if the system is continuously changing. One extended feature of pSearch [113] is to combine the efficiency of DHT systems and accuracy of information retrieval algorithms. Xu, et.al. [114] reports that a perceivable improvement can be gained in a logical routing cost by adopting their proposed algorithms.

However, one requirement in DHT is that all nodes have to be highly coupled. This is one of the critical situations that the future digital libraries have to avoid. In addition, document pointers over the peers may be unevenly distributed and a global state is required beforehand to support the algorithm used in DHT. These can also be too restrictive to meet the autonomy requirements in future digital libraries.

3.4.4 P-Grid

P-Grid [115] is a kind of Semantic Overlay Network (SON) [104], which differs from other approaches such as Chord and CAN, in terms of practi-

cal applicability (especially in respect to dynamic network environments), algorithmic foundations (randomized algorithms with probabilistic guarantees), robustness, and flexibility. The most important properties of P-Grid are: complete decentralization; self-organization; decentralized load balancing; data management functionalities (update); management of dynamic IP addresses and identities; efficient search[115].

3.4.5 HyperCup

HyperCup [116] proposes a graph topology which supports a large number of peers while maintaining relatively low network diameter. An example is shown in Figure 3.7, illustrating a three dimension hypercube. The maximum length between two nodes is 2 while $2^3 = 8$ nodes (ie. peers) are involved.

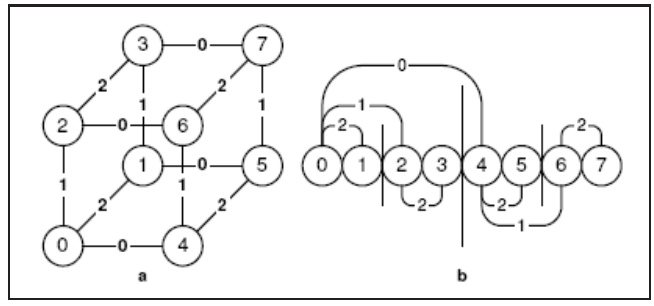

Figure 3.7: Hypercube Graph and Serialization Notation, from [4]

The number of messages generated when peers leave and join the network is $O(log_b N)$ (b is the base of the hypercube), which can be more efficient than DHT algorithm. Moreover, peers can join and leave the self-organizing network at any time, and the network is resilient against failure. Work in [116] reports also that a global ontology can be used to determine the organization of peers in a graph topology, allowing for efficient concept-based search.

3.4.6 Piazza

Piazza [117] is a peer data management system that enables sharing heterogeneous data in a distributed and scalable way. The assumption in this system is that participating peers have similar content to share within

each other. To enable interoperation, pair-wise mappings are defined between their schemas. Since these schemas are constrained in a specific domain, mapping rules are relatively easy to create. Individual peer can thus formulate queries over particular schema. Piazza also creates a query answering system for expanding recursively any mappings relevant to the query, retrieving data from other peers.

3.4.7 JXTA Search Edutella, Bibster

JXTA [118] is a P2P interoperability framework created by Sun Microsystems. All peers can publish their profiles (i.e., content summary) in way of 'advertising'. One peer in JXTA can thus discover other peers by discovering posted 'advertisements' and then join favorite peer groups. Communications between peers are conducted by 'pipes' specifically generated by them. Typical systems include Edutella [34] and Bibster [35]. Both of them support metadata search within P2P networks while the former focuses on educational domain and the latter on bibliographic records respectively. JXTA itself can be regarded as a super-peer network consisted of many 'rendezvous' peers [118]

3.4.8 RDFPeers

RDFPeers [119] is a very interesting approach by extending DHT to support searches over RDF triples. Basically, RDFPeers becomes a scalable distributed RDF repository that stores each triple at three places in a multi-attribute addressable network by applying globally known hash functions to its subject, predicate, and object. Such an approach is very suitable to search through highly distributed RDF repositories.

3.4.9 OAI-P2P

The ongoing OAI-P2P project [33], built atop Edutella, aims to merge the concepts in Open Archive Initiative-Metadata Harvesting Protocol (OAI-PMH) [66] with a true P2P approach, where metadata records on *data providers (DP)* (c.f. [66]) can be harvested by other peers directly. That is, *service provider (SP)* (c.f. [66]) can be bypassed and records end user retrieved finally can be more up-to-date. OAI-P2P made one leap forward allowing different metadata schemas to be published in P2P community. It reuses the *ListMetadataFormats* request in OAI-PMH which enables peers to specify supported metadata schemas, and then applies the query

exchange language (QEL) to rewrite queries. In the current approach, the semantic heterogeneity issue is not substantially dealt with in OAI-P2P though. One part of its future work is to translate between different schemas, eg., from MARC to DC [33].

3.4.10 Semantic Overlay Networks

Another interesting approach currently conducted is the Semantic Overlay Networks (SON) based P2P systems. In SON, a peer typically connects to a small set of random peers (their neighbors), and queries are propagated along these connections. As mentioned previously, query flooding tends to be very expensive. In contrast, connections between peers are influenced by *content*, for example, peers having many "Ang Lee" files will connect to other similar ones. In such a manner a Semantic Overlay Network is formed by these semantically related peers. When a query generated in certain peer, it would be routed to appropriate neighbors, increasing the chances of finding matched files quickly, and reducing the search load on peers that have unrelated content. Evaluation conducted in [104] shows that SONs can significantly improve query performance while at the same time allowing systems to decide what content to put in their computers and to whom to connect.

3.5 Challenges in P2P Networks

It is clearly not wise to conclude that P2P networks can solve all potential problems in current digital library construction, while it is absolutely incorrect as well that client/server models will no longer work for digital library. And it is clearly incorrect that P2P network shall substitute client/server in the near future.

Practically speaking, relationship between these two infrastructure is more complementary, rather than competitive. Most of the time we consider the overall system performances which include flexibility, scalability and adoptability. We list main criteria we consider important in comparing P2P network with client/server systems (cf. Figure 3.1), aiming to have a clear understanding on the advantages and disadvantages of P2P networks.

Additionally, P2P networks deal only with how to access digital libraries freely (ie. a system infrastructure issue), rather than other important issues, such as interoperating among *semantically heterogeneous*

Table 3.1: Advantages and Disadvantages of P2P Systems (Compared with Client/Server Systems)

	P2P System	**Client/Server System**
Cost	less expensive and easy to install and maintain.	more expensive to buy
Independence	high	low
Flexibility	high	low
Scalability	inherent scalability	limited
Reliability	less reliable (eg. home PC can be server)	more reliable (dedicated server)
Security	limited	high levle of security.
Robustness[a]	robust	moderate
Database Application	No good for database applications.	handles shared database applications.
Adoptability	Good for file, printer sharing (simple)	Work for any application

[a]Here it indicates robustness to single-point-failure

metadata dispersed in digital library system.

In this thesis, we focus mainly on different features of P2P network and investigate how to apply them in constructing digital library systems, aiming to share and search heterogeneous metadata records in a seamless manner. Many other critical challenges in P2P networks will not be covered in this thesis, such as security, reliability and data integration, which do not seem to have been fully researched and successfully solved so far.

3.6 Chapter Achievement

- The digital library architecture of the future is expected to support a more robust, scalable and dynamic environment, where cooperation among mutually interested libraries is easy to set up.

- P2P overlay network is one of the methods to facilitate cooperation among digital libraries and improve the accessibility of library services. As against the client/server architecture, P2P overlay networks provide a more open architecture by decentralizing the control

from servers, which in turn improves system robustness and scalability.

- The model of the P2P overlay network is not unique. The type of model is as follows:
  - Pure P2P Model: no centralized controlling node exists, but *rules* can be created in locating peers;
  - Hybird P2P Model: services, such as discovery, lookup and content, are included in this model;
  - Super-peer based Model: an integration of the pure P2P model and the client/server model, which in turn forms a two-tier P2P model
- Existing P2P systems have been surveyed and challenges in P2P networks have been discussed.

Chapter 4

Appropriate P2P Infrastructures for Semantic Search

4.1 Introduction

Based on the survey on existing P2P systems, we move further in this chapter with a summary of these P2P systems and propose a benchmark for choosing appropriate P2P infrastructures for constructing digital library systems under specific requirements. After a revisit on requirements for building future digital libraries, a super-peer based topology is selected as a walk-through example. We are to present problems in classic super-peer based overlay network and propose solutions accordingly. To justify the proposals, several evaluations are to be conducted in the final section.

4.2 Benchmark for Selecting Appropriate P2P Architectures

Basically, many criteria shall be taken into consideration in comparing P2P systems, such as system hardware, communication protocol and security. This thesis work will focus on the issue of information searching and a comparison will be conducted over significant features related to searching. Based on the survey conducted in Chapter 3, we illustrate in Table 4.1 the comparisons over aforementioned P2P systems in aspects of

markup schema, hash table usage, semantic routing style, query forwarding support and semantic query support.

If we are allowed to say that HTML led to information *islands* on the Web, which are only able to be accessed by hyperlinks, the P2P networks brought us just an alternative to cluster mutually interested information islands. These different 'islands' may have different constraints on querying according to various kinds of applications. As to applications which query only few metadata fields, a centralized server-based P2P network (e.g., Napster) is sufficient. In fact, some practical applications, such as Napster using a centralized directory to maintain music files' names; and Gnutella using keyword-based searching over a query flooding P2P environment. Moreover, if libraries can be highly coupled, DHT-based solutions can be used to achieve more efficient and effective performance. One issue which needs further clarification here is that DHT-based solution can only release the impact of frequent requests for some information. It can not release the impact of data *hotspots* due to key collisions which may be caused by too much entities/data being associated with one key [120, 121]. Recent approaches in super-peer based topology [122] or semantic overlay network (SON) [104] can be considered as alternatives to improve efficiency in discovering/locating appropriate peers. These approaches can be contributed for requirements when many digital library systems take *autonomy* as a central value since these approaches can support a more flexible mechanism for loosely coupling among peers. It is dissimilar with the 'rigid infrastructure as in DHT, although the latter makes it easier to locate content later on.

In our approach, more considerations are to be taken into searching in a heterogeneous and distributed environment. Here, a discussion of applying aforementioned different search methods mentioned (c.f. Chapter 2) in varied situations would be necessary. Basically, one search method may be found more suitable than the other in some application scenarios. On one hand, many collections in participating libraries may have various metadata schemas which involve multiple fields, such as title, author and publication. That is, searching over collections can be roughly regarded as a matchmaking procedure on related fields recursively. So, keyword-based search or XML-based search [123] would be effective at this point. In addition, keyword-based search is a kind of full-text based approach which processes the textual information literally while not bothered by the contextual meaning of documents.

On the other hand, there is a critical problem in using keyword-based

Table 4.1: Summary of Typical P2P Systems. From [11]

System	**Markup-Scheme**	**Hash Table Usage**	**Semantic Routing**	**Query Forwarding**	**Semantic Query**
Gnutella	Keyword	No	No	Yes	No
Naspter	Keyword	No	No	No	No
Freenet	Keyword	Yes(binary)	Serial	Yes	No
Routing Indices	Keyword	No	Serial	Yes	No
Chord	Keyword	Yes	Parallel	Yes	No
CAN	Keyword	Yes	Parallel	Yes	No
pSearch	Keyword	Yes	No	Yes[a]	No
P-Grid	Keyword	Dist.Srch.Tree	Serial	Yes	No
HyperCup	Keyword	Yes	Separate HyperCube	Yes	Yes
Piazza	Database	No	No	Yes	Yes
JXTA Search	XML	No	Parallel	Yes	No
Edutella	RDF	No	Parallel	Yes	Yes (regional)
Bibster	RDF/ DAML+OIL	No	Parallel	Yes	Yes (global)
RDFPeers	RDF	Yes	Parallel	Yes	Yes (global)
OAI-P2P	RDF	No	Parallel	Yes	Yes (regional)
SON	Keyword	No	Parallel	Yes	Yes (regional)

[a]IR Query Expansion

approach as well. That is, as long as there are thousands of peers in a P2P system, it would be problematic to collect certain global statistic information, such as inverted document frequency (IDF) if assuming Vector Space Model (VSM) is adopted (c.f. [2]). Even if we can avoid such problems [124, 125], we may still suffer from another problem that a peer would join or leave the system at any time. In this case, the index file or the collected global statistic information (if have) would be out of date and must be updated when such situation happens. Actually, when more and complex metadata elements get involved, such as Bibtex metadata with up to 100 metadata entries [35], it would be inefficient for keyword-based search to go through these entries respectively. So, metadata-based search or ontology-based search would help at this point by supporting more complex queries. Edutella and Bibster, in this concern, demonstrate the possibility to conduct complex queries over metadata records, by using RDF Query Language (RQL) [126] alike query language and RDF-based database management systems [8]. This approach provides us an opportunity to use ontologies to express relations between metadata terms and realize semantic-based search by processing these relations which appear in user queries. Distinct from the keyword-based approach which is not constrained by domain schemas, the metadata and ontology based approaches can not directly search collections created in different schemas without reformulating/rewriting queries on the original schema.

We further discuss in this chapter constructing P2P-based digital library systems. Admittedly, there are a number of requirements to be considered, including system infrastructure (architecture, protocols, syntactic solutions, encoding schemas and identifier systems), standards, organizational and legal matters, supporting services such as different registries and semantic knowledge bases and knowledge organization system (including foundational and core ontologies). However, in this thesis we concentrate on the following critical issues:

- Degree of autonomy: does the library accept arbitrary incoming queries? Or can it support a common shared schema? It is required that queries created in different schema shall be reformulated before sending them to other connected P2P system.
- Keyword-based search or metadata/ontology-based search;
- Multiple (heterogeneous) metadata schemas support: e.g., LOM and DC.

- Metadata records harvesting: if it is not necessary to keep data up-to-date, consistency issue must be considered.
- Authentication: must the library users be authenticated?
- Peer Selection/Discovery: do it need to locate specific libraries or just let system to find them dynamically?

Based on the discussion on Chapter 3, we specify a working benchmark in Table 4.2 illustrating appropriate P2P infrastructures which can be adapted for distributed digital libraries construction.

4.3 A Super-Peer based Network Supporting Federated Sesarch

4.3.1 Requirements

Suppose that a large number of distributed digital libraries to be federated. Advanced search functionality, such as semantic search, is to be supported for searching across distributed collections. In addition, due to the *heterogeneities* in digital libraries, some digital libraries are independent and reluctant to act as a 'client' to other systems, while others are willing to conduct some modification in order to be accessed in a broader view, especially when they have limited resources.

With references to the working benchmark in Table 4.2, a super-peer based infrastructure or semantic overlay network can be is selected to deal with the application scenario. The major deterministic reasons are as follows:

- Scalability: it requires that the system can cope with the *bottleneck* problem, or at least realize 'multi-points of failure' instead of 'single point of failure', which is also critical for system robustness;
- Flexibility on the role of digital library: autonomous nodes can act as super peers while weak nodes may trade their independences to reach an agreement with others and act as clients to a specific super peer;
- Complex search capability: due to that super peers are autonomous, they are able to support complex search as a standalone server, besides that they have to maintain a list for neighboring peers.

Table 4.2: Working Benchmark for Selecting P2P Infrastructures for Digital Libraries

Scale	**Metadata records**	**Semantic support**	**Autonomy**	**Adaptable P2P Network**	**Info. Srching Technique**
small	few	No	high	pure P2P, RI	Information retrieval (IR)
small	few	No	low	pure P2P, Central server-based P2P, DHT	IR
small	many	No	high	pure P2P, RI	XML-based IR, RDF database
small	many	Yes	high	pure P2P, RI	RDF database
small	many	No	low	pure P2P, Central server-based P2P	XML-based IR, database
large	few	No	high	Super-Peer, SON	IR
large	few	No	low	DHT, Central server-based P2P	IR
large	many	No	high/low	Super-Peer, SON	XML-based IR, database
large	many	Yes	high	Super-Peer, SON	RDF database + RQL
large	many	Yes	low	Super-Peer, SON, DHT + logical layer	RDF database + RQL

In order to have a thorough and comparable study, the classic super-peer system model is revisited in the next section.

4.3.2 Classic Super-Peer System Model - Revisited

To make it consistent, we bind the understanding of super-peer network to the definition in [122]. According to Yang[122], *super-peer networks* present a cross between pure and hybrid systems. A *super peer* (c.f. Figure 3.6) acts as a central server to a subset of clients. Clients submit queries to corresponding super peer and receive results from it. Basically, connections among super peers form a *pure* P2P system, and super peers are responsible for submitting and answering queries on behalf of client peers and themselves.

To make it concise, later on we call a super-peer and its clients a *cluster*, where *cluster size* is the number of nodes in the cluster, including the super-peer itself. In an extreme condition, a pure P2P network can be regarded a regressive super-peer network where cluster size is 1 - every node is a super peer with no clients.

Applying super-peer networks to digital library construction, we can regard a cluster as a set of digital libraries where one of them is selected or simply agreed by other libraries as a super node. To maintain such a cluster, this super node can keep an index over its clients' data or simply provide a lookup service. In file sharing applications, the super peer may just keep inverted lists over the titles of files owned by its clients. If the super peer finds any results, it will return with message including results and the address of clients. But in a more advanced situation where it is impractical or expensive to keep a central index, e.g., advanced search requirements on multiple metadata fields or SQL-alike search, the super peer evolves into a 'lookup' server providing coordinating services for its clients. When a super peer receives a query from a client, it will submit the query to its neighboring super peers (*neighbors* for brevity in the rest paragraph) as if it were its own query, and it will forward any returned messages back to the requesting client. That is, outside the cluster, a client's query will not be distinguishable from a super peer's query. The advantage of this mechanism is that clients are saved from processing any extra queries and network traffic, so weak nodes (e.g. nodes having limited network bandwidth) could act as clients, while strong nodes can be united into an efficient network.

In contrast to client/server architecture, super-peer network allows de-

centralized networks to run more efficiently by distributing load to nodes that can handle the burden. With the capability of allowing multiple separate points of failure, super-peer network increases the robustness of digital library systems. On the other hand, when the size of the network soars, in contrast to pure P2P network which may suffer deteriorated performance, e.g. slower response time or fewer available sources, super-peer network can take the advantage of heterogeneity (i.e. mutually exclusive roles: client peer and super peer), assigning greater responsibility to those who are more capable of handling query answerings.

However, extreme scalability and dynamism pose a problem for a generic super peer network. In an extreme unbalanced situation, there is a high possibility that super peers are overloaded. Furthermore, when a super peer fails or simply leaves, all its client peers will be disconnected from the network until they can find a new cluster to connect to. Although reliability can be increased by introducing redundancy, e.g. k super peers instead of one super peer, into the design of the super peer [122], such solution comes at a cost. Inside a cluster, there is an extra cost of maintaining clients' information on super peers and the communication among them would be k times greater than before. Also, the communication cost with neighbors is high since all super peers in a cluster may receive messages from neighbors. Hence, in this thesis, instead of applying super peer redundancy, we come up with an enhanced super-peer model to alleviate previous mentioned problems.

4.3.3 Enhanced Super-Peer Model for Federated Search

To cope with the practical application, we consider a federated digital library system $\mathcal{DL}$ consisting of a large collection of *nodes* which exceed the capability of conventional client/server model. We assume that all nodes in $\mathcal{DL}$, both client peer and super peer, can be located via unique identifiers. Also, $\mathcal{DL}$ is highly dynamic that new nodes may join at any time and existing nodes may come down or simply leave. Furthermore, a super-peer's capability is limited in contrast to the large scale of $\mathcal{DL}$, so an individual super peer can only connect to a constricted number of neighbors. To extend the capabilities of the super-peer model (c.f. Figure 3.6) described in [122], we make the following enhancements.

Super-Peer Network Initialization

In many research projects, often are super-peer networks generated 'artificially'. For example, in [122] a graph-based topology is set up, where each of them represents actually a single cluster. These nodes are transformed into individual super peer and a number of clients are added to each super peer. The number of clients in a cluster follows the normal distribution $N(u_c, 0.2u_c)$, where u_c is the mean cluster size. However, the actual size of a cluster is decided in a 'bootstrapping' manner since it is a progressive procedure for client peers to discover super peers. Similar to [122], we assume also that there are initially a large number of average nodes $\mathcal{N}$ where no 'role' (i.e. client or super peer) has been assigned to each node yet. But as different from the approach in [122], we allow each node to pro-actively discover existing super peer to connect to, or declare itself as a super peer if it can not find super peer but has capability to accept incoming peers. In system $\mathcal{DL}$, initially there is no super peer available, so only those who have greater capability have the chances to become a super peer.

The rationality behind this extension is that: in practical situation, different peers may have varied computing capabilities or resources, so it is necessary to take into consideration the heterogeneity of peers. That is, weak nodes can be made into clients, while the core of the system can run efficiently on a network of 'strong' super peers. It is also in much closer to practical applications where peers' computing capabilities are varied.

Load Balancing

Load balancing is especially important for the super-peer based networks where it's difficult to predict the number of new client peers that may connect to a super peer or the number of requests that will be issued to the super peer. Consider the situation of peer joining, we observe that the actual number of clients in a cluster is determined by the maximum capacity c_s a super peer s can afford. Generally, c_s is determined by peer's computing capability, storage and bandwidth. Hence, some super peers may have greater capabilities to accept new client peers, while other super peers can only host limited number of incoming peers. Clearly, in the latter situation, if the load can not be alleviated from the weaker super peers, performance bottlenecks (i.e. overloads) may occur in these nodes.

To alleviate such problem, we propose a two-step load balancing mechanism which is applied to distribute loads *dynamically* between a super

peer and its neighbors. In the first step, when new nodes are added to the network, we initialize them all as super peers such that gossiping protocol can be reused for the reconnections with other super peers. Similar to the initialization, peers with lower capacities can be merged to a cluster led by a super peer with greater capacity. The next step which is also the critical one, is transparent to external observers. Generally, if a super peer can not accept a new client any more, it may *push* such client peer to a neighbor whose current load does not reach the limit. On the contrary, 'strong' super peers who can afford greater load can also proactively *pull* client peers to their clusters. Behind the *push-and-pull* mechanism, a list of unclustered peers u_c act as a *coordinator* to accept 'abandoned' but active peers. As soon as a client node is added to a proper cluster, it is removed from the list.

Self-Organizing

The model in [122] does not allow one client peer to communicate with other super peers except the one in its cluster. Under this assumption, if a super peer fails or leaves, all clients connected to it will be lost. To increase the robustness, we introduce the idea of self-organizing to cope with the situation when super peers are unavailable. Basically, first we assume that all nodes (i.e. both super and client peers) in our prototype have unique identifiers, as similar to that in Internet, which are used to locate individual nodes. Second, $\mathcal{DL}$ allows client or super peers sending messages to each other to check whether they are still online. Hence, if a super peer becomes unavailable, all client peers connected will be added into a 'unclustered' list. Each peer in this list, following the strategy of *Load Balancing*, can be assigned to particular clusters; or, simply declare themselves as a *new* super peer which in turn can accept connections from other client peers and finally lead to a new cluster.

4.4 Evaluation

4.4.1 Evaluation Setting

To validate our approaches, we have conducted numerous *simulation*-based experiments. The reason for applying a simulation-based approach is that: P2P systems are usually extremely large scale and dynamic systems where nodes may join and leave freely. However, experiments in a

Table 4.3: Configuration parameters and default values

Name	Default	Description
Super-peer Network Initialization	Gossiping protocol	The way how super-peer network is generated (the intial status).
Network Size	10^5	The number of peers in the network
Maximum Outdegree	30	The number of neighbors
Cluster size	power law distribution	The sizes of clusters are in a power law distribution
Max Capacity c_{max}	100	The maximum capacity of a peer to host client peers
Min Capacity c_{min}	1	The minimum capacity of a super peer to host client peer

practical environment, e.g. thousands of nodes, turn out to be not an easy work at all. In addition, by applying simulation based approaches, it is convenient to set up different application scenarios to study P2P systems' behaviors which are of great interests to us.

A number of experiments have been conducted in this work in order to investigate the performance of different *configuration* of systems. Default configurations defined by a set of parameters are shown in Table 4.3. We are to explain these parameters in detail as they appear later in the section.

The configuration parameters in Table 4.3 describe the desired topology of the network. Unlike [122] where client peers are specifically added to a cluster, we generate a 10^5 size network by gossiping protocol [105]. The gossip algorithm [127] allows a node communicates with a randomly chosen neighbor and exchanges information (eg. computation capacity in our experiment, cf. Table 4.3) with it. One of the advantages of gossiping protocol is that it can be used for computation and information exchange in an *arbitrarily* connected network of nodes. In our evaluation settings, each node is initialized as a super peer and randomly connected with each other. This scenario is the most 'artificial' one in this experiment but it provides us a mechanism for bootstrapping the overlay network in a natural way. In a straightforward way each node in this network is assigned a maximum *out-degree* of neighbors it connects to. A super peer is selected from neighbors randomly provided that the neighbor's capacity is greater than the current local node. Note that the network topology follows power-laws [128], so without losing the generalization, we generate peers' capacities by using a power-law distribution $P(x) = \beta x^{-\alpha}$, where

x is for the capacity of a peer in a range of $[c_{min}, c_{max}]$ (i.e.$[1, 100]$ in our settings), $P(x)$ is the probability of having specific capacity x, and β and α are constant parameters. For simplicity, we herein define $\beta = 1$ and $\alpha = 2$. Observe that although one node with maximum capacity may not be willing to host more clients, we thus collect qualified neighboring nodes (i.e. having greater capabilities) and randomly select one from them. From certain perspective, it is also in line with the random procedure of generating super peers.

In addition, all experiments are conducted in a *round-driven* manner in order to capture system snapshots in specific situations.

4.4.2 Experiment 1 - Super-peer Network Generation via Gossiping protocol

Figure 4.1 illustrates a network of 10^5 size. Ten individual experiments are conducted with out-degrees varying from 10 to 100 but with peer's maximum capacity fixed (i.e. 100). The curves represent the change of the number of super peers in the network after specific rounds.

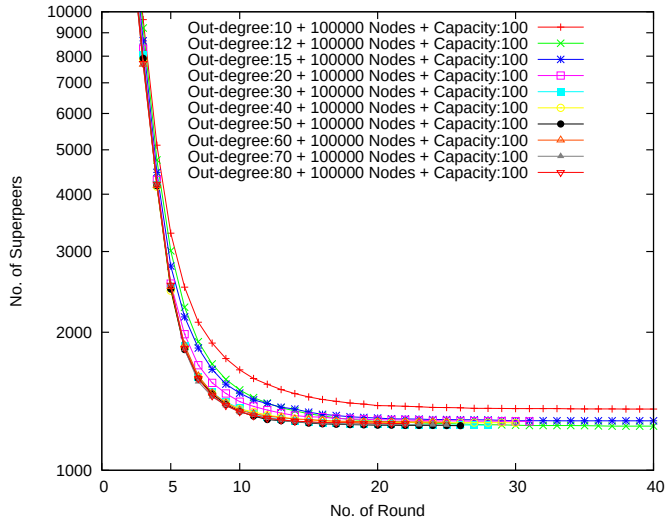

Figure 4.1: Super-Peer Network Generated by Gossiping Protocol(Outdegree)

The value of out-degree indicates a peer's partial view on the network (i.e. its neighbors). The gossiping protocol is adapted in initializing the

super-peer network. At round *zero*, super peers are initialized to connect to their neighbors via their partial views. Then, the bootstrapping step is conducted by a custom-developed super peer protocol where connections between super peers are limited by corresponding *capabilities*. Intuitively, networks should vary significantly by different out-degree parameters, but it turns out in Figure 4.1 that the fluctuations on the numbers of super peers are relatively small. The reason is that in the bootstrapping the decisive factor, the peer's capacity, determines how many neighbors a super peer can connect to. To justify our judgments, we then fix the out-degree parameter and assign different values for peer's maximum capacities, namely, 100, 200 and 500. The results are shown in Figure 4.2.

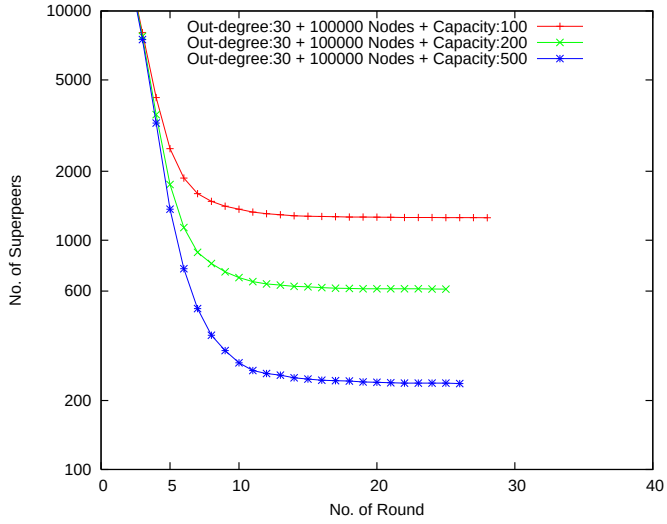

Figure 4.2: Super-Peer Network Generated by Gossiping Protocol (Capacity)

In Figure 4.2 we see that the variations of the number of super peers are in the same pace with peer's max capacity. The larger the capacity, the fewer super peers will be needed, and the less communication overhead among super peers will be. In addition, both Figure 4.1 and Figure 4.2 illustrate that the number of super peers decreases logarithmically before reaching a comparatively stable situation. Interestingly, the time (i.e. rounds) required to reach a stable situation, i.e. with relatively constant number of super peers, is between 10 and 20. This is due to the fact that no super peers should be removed from the network if an approximate

optimized situation is reached, i.e., all client peers are connected to a super peer respectively and super peers are not overloaded.

4.4.3 Experiment 2 - Load Balancing

To evaluate the system performance under increasing load, we set up an experiment where 1000 new nodes are added to the network continuously from round 25 to round 35. The reason for choosing a round 25 is that at that moment the network is to have a relatively stable optimal status. If new nodes are added, the system will churn out.

Figure 4.3 illustrates that the number of super peers increases sharply as new nodes are added. It is because we initially assign the role of 'super peer' to all new nodes and adjust them in later rounds. As shown in Figure 4.3 as well, an obvious decrease happens after round 35, when no new nodes are added into the system. Although the system comes to a stable situation finally (c.f. after round 50), the number of super peer is slightly larger than the system which has a fixed number of nodes.

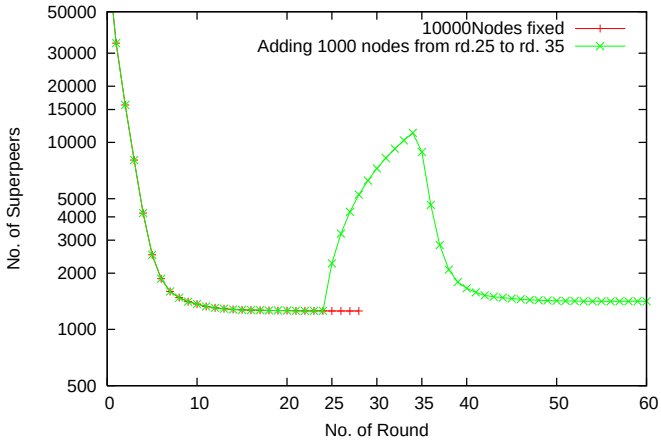

Figure 4.3: Load Balancing (1000 Nodes join the network each time from round 25 to round 35)

4.4.4 Experiment 3 - Self Organizing

The self-organizing functionality we aim to achieve in this work is to dynamically adapt to node failure and degradation, and react to changes

caused by them. The general strategy below is applied for nodes acting different roles.

- Super peer: it maintains only neighboring super peers' information without worrying about the existence of its client peers. Periodically it checks the connection status of its neighbors and removes those unreachable.

- Client peer: it periodically checks whether its super peer is online, especially before it sends requests. If the super peer is down, it searches nodes (i.e. super peer) which have more capacity and may join such a cluster led by the new super peer. Otherwise, if no super peer wants to host it, it may finally declare itself as a super peer, waiting for incoming client peers.

Two kinds of scenarios are investigated in this work. The first scenario is that peers keep leaving while the second one is that disastrous failures happen in peers, e.g., half of the peers in the network are crashed. Results are shown in Figure 4.4 and Figure 4.5 respectively.

In the first scenario, we force 500 peers to leave the network from round 10 to 90 continuously. Figure 4.4 illustrates the changes in the number super peers follow a undulant manner due to the self-adjustment each time. As we can see from the enlarged figure, the tendency in the number of super peers is decreasing along with the running rounds, as is in accordance with the fact that the total number of peers is reduced.

In the experiment illustrating effects brought by catastrophic failure in peers, we assume half of the peers in the network are crashed in round 5, 10 and 20 (c.f. Figure 4.5), which are at the beginning, middle and end of the procedure for the network reaching optimal status (c.f. the 'no leaving' curve in Figure 4.5).

Figure 4.5 shows that no matter when such a catastrophe happens, a transient phenomena that the number of super peers may surge to around 25000. The reason is that at that moment a large number of client peers which lost connection to their super peers have the chances to declare themselves as super peers, especially when they can not find neighboring super peers who are willing to accept them. In fact the system is actually sparsely connected as in the initialization status. However, after the stumbling, the system heals itself quickly to an optimal status after 15 rounds, which justifies that merging between these newly generated super peers helps reduce the number of total super peers.

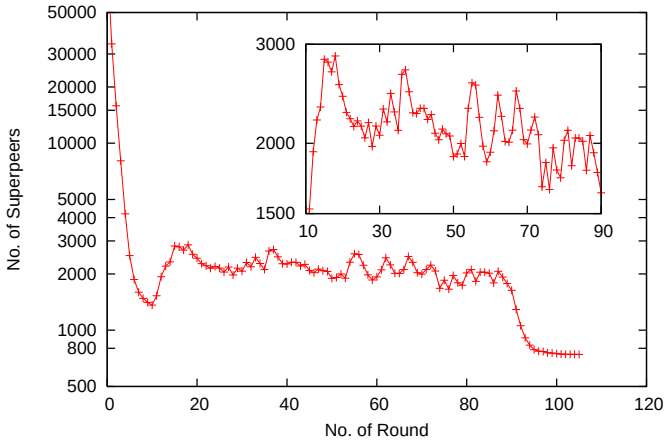

Figure 4.4: Self-Organizing in Scenario of Continuous Peer Leaving

Results illustrated in Figure 4.4 and Figure 4.5 prove that the protocol helps the system adapt dynamically to continuous peer's failure (e.g. peer leaving) and huge degradation and changes in the network.

4.4.5 Summary

Super-peer based network shares features embodied by both pure peer-to-peer networks and client/server systems, and thus can be employed in information sharing applications which may host heterogeneous entities. The work conducted by Yang, et.al. [122] has discussed the general issues in the design of super-peer overlay networks. However, mechanisms have not yet been investigated for generating super-peer networks in a bootstrapping manner and supporting self-organizing capabilities.

The first commercial system employing super-peer network is perhaps Kazaa [107] which is devoted for file-sharing applications. However, no public report is available concerning the protocols and system performances.

JXTA [118], a P2P interoperability framework created by Sun Microsystems, has been used in sharing distribution information resources. In JXTA, communications between peers are conducted by pipes generated specifically by themselves. Typical systems include Edutella [34] and Bibster [35]. Both of them support metadata search within P2P networks while the former focuses on educational domain and the latter on biblio-

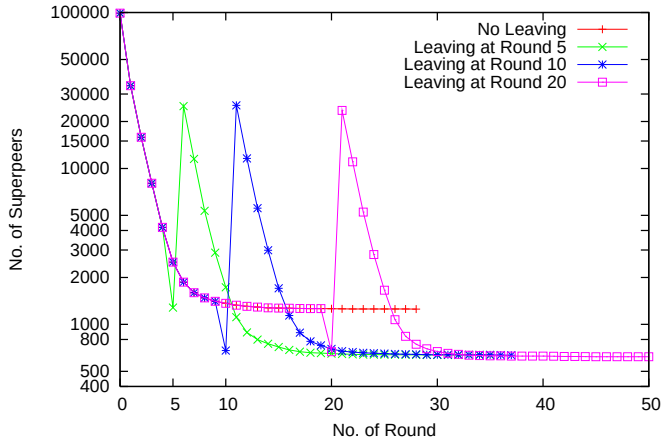

Figure 4.5: Catastrophe Recovery (50000 Nodes left in round 5)

graphic records respectively.

Peer-to-Peer (P2P) overlay network becomes a substantial research topic in building future digital libraries. However, due to the obscurity in the system behaviors by applying the protocols, considerable efforts are required to investigate such possible performances and propose solutions for implicit problems. In this chapter, we have presented the urgency of exploiting advanced infrastructure for future digital library constructions and described the basic requirements, i.e. system bottleneck and dependences. Among various solutions, super-peer based network turns out to be one of the appropriate solutions [124]. However, due to that classic super-peer networks may still suffer from the problems caused by system's extreme scale and dynamism. Thus, we proposed an enhanced model concerning the initialization of super-peer networks and strategies for load-balancing and self-organizing. Evaluation results show that the enhancements are feasible and can be applied in large scale super-peer overlay networks. Further works include supporting complex search in super-peer based digital library systems since semantic search or complicated metadata search require more advanced processing on individual information collections, as in contrast to simple file sharing which put overhead of maintaining an index at the super peer.

4.5 Chapter Achievement

- Proposed a tentative benchmark for choosing appropriate P2P overlay networks in specific application scenarios.
- Presented a walk-through example, illustrating how to choose candidate P2P networks.
- Discussed problems in classic super-peer based overlay network and propose solutions, such as load-balancing and self-organizing.
- Conducted extensive evaluations for justifying the applicability of the enhanced super-peer model.

Chapter 5

Metadata Heterogeneity

5.1 Introduction

To facilitate searching across distributed digital libraries for relevant records created in heterogeneous schemas, two approaches are required: 1) building a scalable and flexible system infrastructure; 2) generating a mechanism for metadata interoperation. Chapter 3 will justify the possibility of applying P2P-based overlay network to provide access freely to *distributed* and *autonomous* digital library systems, while Chapter 6 describes potential effects of Semantic Web technologies on the digital libraries in a manner of making resources more understandable to machines. This chapter is devoted to investigating the metadata heterogeneities in digital library applications, with more emphasis to be placed on the semantic heterogeneity issue because sophisticated solutions to syntactic and structural issues have been well developed.

5.2 Heterogeneity Categories

The issue of data heterogeneity has long been well recognized in the database community (c.f. [18, 19, 20, 21, 22, 23]). In general, heterogeneity issues can be divided into three categories:

- **Syntactic heterogeneity**: it is concerned with the "standards" involved in the effective "communication, transport, storage and representation" of metadata and other types of information [129], e.g., heterogeneities in metadata formats, query languages.

- **Structural heterogeneity**: it emerges when sources adopt different data models, data structures or schemas. For examples, relational and object-oriented database models.
- **Semantic heterogeneity**: it is due to the semantic conflicts in terms, phrases, etc, which are adopted by different metadata schemas but actually expressed in various ways.

Throughout this thesis we will mainly focus on the issue of semantic interoperability, while describe briefly current solutions for structural heterogeneity.

5.2.1 Integrating Syntactically Heterogeneous Sources

As mentioned previously, syntactic heterogeneity is caused by discrepancies across the different protocols or languages to describe data, or to query, analyze and update them. In digital libraries, syntactic heterogeneities may prevent sharing (and hence integration) even if there are no infrastructural barriers. For example, two digital libraries may use relational databases, but with slightly different dialects of the language (e.g., SQL) used for querying. Information sharing between digital libraries is mostly processed in form of metadata, such as MARC, however, it does not mean these metadata standards also cover internal data storage format. Actually, specific protocols, e.g. Z39.50 protocol, have to be supported so as to facilitate information searching. The protocols act basically as a common layer atop information resources in heterogeneous syntax.

Accompanied with the growth of the Web, many digital libraries have built their Web portals, providing services like Web-based searching. Under such prerequisite, XML can be used a 'format' for direct information exchange. The major contribution of XML is its easiness to implement and the uniform data output format, which relieve researchers from creating various data transformation tools. Probed from current projects and systems, it is safe to say that XML is to be a widely accepted standard. Due to the focus of this thesis, we are not to investigate further in this direction.

5.2.2 Integrating Structurally(Schematically) Heterogeneous Sources

Structural (schematic) heterogeneity originates from the difference in data models or schemas, namely, the class hierarchies and attribute structure.

Examples are: particular feature is classified under different object classes in different databases; or an object in one database turns out to be an attribute in another. Actually, the issue of schema heterogeneity has been studied in the database community in recent 30 years and many approaches have been conducted in this field. Herein, we briefly introduce three distinguished approaches in the following:

Global-as-View (GAV)

In the Global-as-View approach, every entity in the global schema is defined as a view over the different source schemas that are to be integrated. A major advantage of the GAV approach is that query answerings are relatively straightforward by referencing to the global schema. That is, incoming queries can be easily expanded/rewritten in the terms used in each local source. As illustrated in Figure 5.1, a global schema $G(A1, X.A2, B1, Y.A2)$ is generated by summarizing sources schemas from X and Y. All entities from source schemas have corresponding names in the global schema, even some of them share the same meaning, such as $X.A2$ and $Y.A2$). However, it also leads to a difficulty in updating global schema because of the dependency between the global schema and the local sources. For example, if the global schema has been updated (e.g. new entities are added), all local nodes have to update their local views on the new global schema. On the other hand, adding or removing sources may result in considerable changes to the global schema. As illustrated in Figure 5.1, if a new node Z has been added to the system, correspondingly the global schema has to be updated into $G'(A1, X.A2, B1, Y.A2, Z.A1, C2)$.

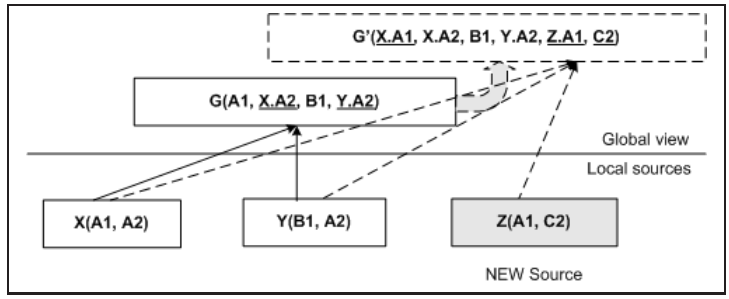

Figure 5.1: A GAV Example

One example applying GAV approach is the TSIMMIS project, which supports rapid integration of heterogeneous information sources that may

include both structured and semistructured data[69].

Local-as-View (LAV)

As contrast to the Global-as-View approach, in Local-as-View approach source views are used in exactly the opposite way. These views define how local information maps to the global schema by expressing a mapping from each relation in the local schema to a (set of) relation(s) in the global schema [19]. An LAV example is shown in Figure 5.2, as compared to GAV-based approach. The main advantage of the LAV approach over GAV approach is that there is no dependency on global schema. In LAV, each source schema is mapped to the global schema. Adding new sources to system requires only definitions of necessary mappings between the source schema and the global one. However, in this approach query answering becomes more difficult because query reformulation is difficult to conduct, and instead an 'abduction-like' approach is required [18].

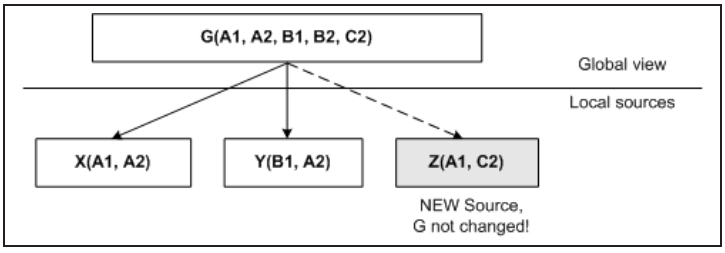

Figure 5.2: A LAV Example

Gloabl-and-Local-as-View (GLAV)

The GLAV approach, in a more reconcilable way, has combined the expressive powers of both LAV and GAV [130]. In GLAV approach, the independence of a global schema, the maintenance to accommodate new sources, and the query-reformulation complexity are the same as in LAV. However, GLAV can create a view over sources by generating a view over global schema described by source descriptions(c.f. G in Figure 5.3). Hence, GLAV can derive data using views over source schemas, which is more expressive than LAV's capabilities. On the other hand, it allows reformulation on global schema (i.e. conjunction), which is beyond the expressive ability of GAV. The G' in Figure 5.3 is just the conjunction of G and the schema of new node W.

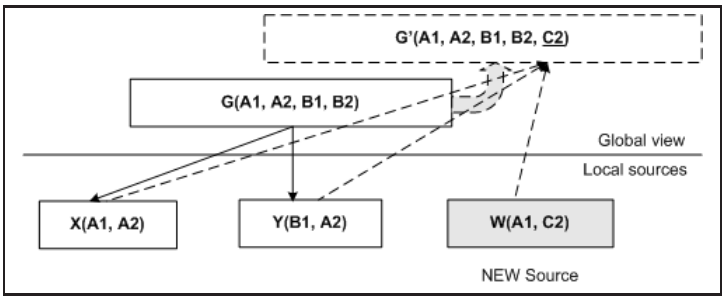

Figure 5.3: A LAV Example

These approaches work for many specialized applications very well *but* are problematic in decentralized and highly dynamic environment. Basically, view-based methods facilitate users to access to a collection of related data sources which can somehow constitute a single data source (e.g., a 'one-stop' search facility). To achieve this, an integrated global 'view' of the data schemas in all the sources is provided. Hence, users in these systems have to formulate queries over a global view instead of dealing with individual data sources. Actually, a global schema is assumed *by default* in data integration systems to deal with heterogeneous data sources [17], however, such an assumption is hard to realize in some P2P systems, as peers (digital library nodes) may join or leave for several reasons like network bottlenecks and maintenance problems. Therefore, soliciting all peers to stand to a commonly agreement is rather impractical. In other words, **instead of a *unique* global schema, *multiple* data schemas may exist in P2P-based systems**. Here, one direct task is to *reformulate* queries in corresponding format used in other peers. To face such tasks, there is a possible challenge in generating semantic relations between elements in heterogeneous metadata [131][132], so as to make the reformulation justifiable.

5.2.3 Integrating Semantically Heterogeneous Sources

Semantic metadata interoperation is not a new issue in not only digital library community, but also in a wider world, like Web. Over the last two decades, different approaches have been conducted to achieve terminological interoperation between heterogeneous metadata schemas, such as automatic (semi-automatic) classification, crosswalks and mappings and controlled vocabularies. In addition, RDF and XML have been widely

accepted as an easily processable and machine-readable format. Herein, we introduce briefly applicable methodologies adopted in digital library applications.

Union Catalog

A union catalog contains records about materials in several collections or libraries[15]. Often such union catalog is maintained in a highly centralized model. In fact, various forms of union catalog have been applied in the library community for over a century. The union catalog enables multi-site libraries to share bibliographic data, user data, and reader services while preserving the degree of autonomy required by each local member. From the users' perspectives, it is in fact a "one-stop" portal which enables them to search across multiple libraries. Users can just specify the libraries they want to query, without being bothered to search across all these libraries individually.

Strictly speaking, union catalog is a Global-as-View approach, but rather a semantic-based approach since it simply combines various schemas into a specific portal. We put it here mainly because of the possibility of searching across *semantically* heterogeneous digital libraries by using union catalog. One example as predicted by Leazer [133] is to include in catalog information on derivative bibliographic relationships for an accurate control of bibliographic records.

Crosswalk/Mapping

Crosswalk/Mapping is a method dealing directly with the meanings of metadata elements. As remarked by Godby[134], crosswalks are a mapping table of equivalences used to 'translate' between different metadata element sets. In other words, elements in one metadata set are correlated with elements in another that have the same or similar meanings. A exemplar crosswalk from MARC to unqualified Dublin Core is shown in Table 5.1:

A general algorithm in creating such mapping tables is depicted in Algorithm 1:

Crosswalks evolve from a demand for compatibility with heterogeneous collection since relevant records may be created in distinct metadata formats. Mapping tables can thus be used to facilitate federated search by rewriting queries in original schemas to queries in the target one. Although

Table 5.1: MARC to unqulaified Dublin Core Crosswalk, from [12]

DC Element	**MARC Fields**
title	245
creator	100, 110, 111, 700, 710, 711,720
subject	600, 610, 611, 630, 650, 653
description	500-599, except 506, 530, 540, 546
contributor	-
publisher	260ab
date	260$c
type	655, etc
format	856$q
identifier	856$u
source	786ot
language	008/35-37,546
relation	530, 760-787ot
coverage	651, 752
rights	506, 540

Data: Source schema S and target schema T
Result: mapping table M from S to T
initialization;
make a list of all elements $(e_{s1}, e_{s2}, ..., e_{sm})(m = \|S\|)$ in S ;
read e_{s1} ;
$e_{si} = e_{s1}$;
while $e_{si}(i \in m, m = \|S\|)$ *is not the last element in* S **do**
  if *Found matched element* $e_{tj}(j \in n, n = \|T\|)$ *in* T. **then**
    Store (e_{si}, e_{tj}) into table M ;
  end
  else
    Store $(e_{si}, NULL)$ into table M ;
  end
  read next element in S ;
end

Algorithm 1: 2D Mapping Table Generation Algorithms

crosswalks are useful for "promoting some degree of interoperability" [134], they may lead to null or inexact correspondences (c.f. Figure 5.1). Indeed, many crosswalks provide only one-to-one mapping, but mappings for one-to-many and many-to-one can not be handled well all the time [135]. Besides, loss of data may also be found in 'one-way' mappings, especially in mapping from a complicated schema to a simple one because corresponding terms may not be found.

Application Profiles

Metadata interoperability is a fundamental requirement for access to heterogeneous digital libraries, as well as information on the Internet. The interoperability between metadata standards is particularly essential in situations when a single query is expressed multiple descriptive formats, or relations have to be set up between different metadata standards. Often there is a dilemma in using *simple* metadata standards, e.g., Dublin Core, which has been applied largely in annotating bibliographic records, but cannot satisfy the requirements of particular communities. One example as described by Hunter and Lagoze[136] is that standards such as TV-Anytime[137] , MPEG-21[138], BIBLINK[139] and OAI[66] need to combine metadata standards for simple resource discovery (DC), rights management (INDECS[140]), multimedia (MPEG-7[141]), geospatial (FGDC[142]), educational (GEM[143], IEEE LOM[144]) and museum (CIDOC CRM[31]) content, to satisfy their application-specific requirements.

In order to enable flexible, dynamic mapping between complex metadata descriptions which mix elements from multiple domains, *application profiles* is created. According to Heery and Patel [145], application profiles is the schemas which consist of data elements drawn from one or more namespaces, combined together by implementers, and optimized for a particular local application. Adhered to this definition, this approach is to accommodate individual needs and elements in standard or commonly shared schemas are adapted to cater to local specific needs. Thus, elements from distinct metadata sets can be syndicated according to particular application profiles. So far approaches to application profiles have been based on either RDF Schema[146] or XML Schemas[147]. One example adopting a pure RDF Schema approach can be found at the SCHEMAS project[148]. Another interesting approach led by Hunter and Lagoze[136] is combining XML Schema and RDF Schema to fit into the overall web metadata architecture. They also demonstrate how interoperability be-

tween application profiles can be enhanced by using such dual schema approach.

Registries

According to [149], the term "registry" covers a broad range of databases, documentation services, or Web-based portals providing access to schemas. Generally, metadata registry is composed of an index of officially defined metadata terms, which can then be extended by local particular domains, services or projects. In another word, users/communities can enable the reuse of existing (registered) elements rather than reinventing their own. Therefore, it is safe to say that registering profiles can help to harmonize metadata usage in particular domains. Some practical projects are: the Schemas project which aims at providing a selected and annotated overview of metadata vocabularies and their usages in application environments [148]; the EU Cores project which includes registry of core vocabularies and profiles and developed a schema creation tool and Web interface to register schemas [150]; and the well-known Dublin Core Metadata Registry, which turns out to be authoritative source for DC and promote the discovery and reuse of exiting metadata definitions [151]. Baker [149] also concludes that "almost universally, registries are seen as our best hope in the medium term for a scalable solution to the problem of mapping and translating between a diversity of schemas".

Others - Derivation, Satellite and Switching

Other interesting methods for interoperating among metadata schemas are *Derivation*, *Satellite and Leaf Node Linking*, and *Switching* [135] [152]. The derivation approach is conducted by developing a specialized or simpler vocabulary with an existing, more comprehensive vocabulary as a starting point or model. The linking method aims at creating a list of terms linked with other terms (e.g. in a form of hierarchical 'satellite') that may not be conceptual Roughly, these three methods can be regarded as special cases of *mapping* respectively. For example, the method of derivation generally develops simpler schema (terms) from an initial but more comprehensive one, as such as direct *mapping* from a complex schema to its simplified version. The satellite and leaf node linking, literally, requires *mapping* from broader terms in satellite of superstructure to narrower terms in the leaf level. Similarly, switching requires all cooper-

ating schemas to translate (map) their terms to an intermediary schema.

5.2.4 Metadata Encoding Methods

Obviously, any kinds of metadata records shall be encoded in certain way to facilitate preservation as well as interoperation. Here we introduce two prominent candidates as below.

RDF/OWL

RDF[153] and OWL[154], as well as XML[155] (only in concern of structure), are gaining the popularity of encoding metadata records. One example is the SIMILE project[156] which aims to leverage and expand DSpace. In SIMILE, RDF and OWL are used to enhance inter-operability among digital assets, schemas, metadata, and services that are distributed across individual, community, and institutional stores. Besides the general capability of describing resources, RDF (as well as OWL) holds other critical characteristics, such as *Independence* and *Interchange* [157]. On one hand, *Independence* implies any independent organization (or even person) can invent *Property* which is generally inherited from some metadata elements. For instance, one can use *Author* in a publication site, and another one can use *Director* when associated with movies. And both of them, to be more general, can also be represented as *Creator*. From the semantic perspective, the feature of *Independence* does not refrain us from using flexible terms to describe similar concepts. On the other hand, *Interchange* reveals the fact that RDF can support the exchange of information and knowledge on the Web or large scale Intranet environment, since RDF Statements can be converted into XML, and provide a decent way to represent entities (nodes) and relationships. Further descriptions on XML, RDF and OWL are in Section 6.3.3.

Metadata Encoding and Transmission Standard (METS)

The Metadata Encoding and Transmission Standard (METS) schema is a standard for encoding descriptive, administrative, and structural metadata regarding objects within a digital library, expressed using the XML [155]. According to [158], a METS document consists of seven major sections as below. A more detailed explanation of each section and their inter-relations can be found in [159].

- **METS Header** - The METS Header contains metadata describing the METS document itself, including such information as creator and editor, etc..

- **Descriptive Metadata** - The descriptive metadata section may point to descriptive metadata external to the METS document (e.g., a MARC record in an OPAC or an EAD finding aid maintained on a WWW server), or contain internally embedded descriptive metadata, or both. Multiple instances of both external and internal descriptive metadata may be included in the descriptive metadata section.

- **Administrative Metadata** - The administrative metadata section provides information regarding how the files were created and stored, intellectual property rights, metadata regarding the original source object from which the digital library object derives, and information regarding the provenance of the files comprising the digital library object (i.e., master/derivative file relationships, and migration/transformation information). As with descriptive metadata, administrative metadata may be either external to the METS document or encoded internally.

- **File Section** - The file section lists all files containing content which comprise the electronic versions of the digital object. *¡file¿* elements may be grouped within *¡fileGrp¿* elements, to provide for subdividing the files by object version.

- **Structural Map** - The structural map is the heart of a METS document. It outlines a hierarchical structure for the digital library object, and links the elements of that structure to content files and metadata that pertain to each element.

- **Structural Links** - The Structural Links section of METS allows METS creators to record the existence of hyperlinks between nodes in the hierarchy outlined in the Structural Map. This is of particular value in using METS to archive Websites.

- **Behavior** - A behavior section can be used to associate executable behaviors with content in the METS object. Each behavior within a behavior section has an interface definition element that represents

an abstract definition of the set of behaviors represented by a particular behavior section. Each behavior also has a mechanism element which identifies a module of executable code that implements and runs the behaviors defined abstractly by the interface definition.

5.3 The Needs for Explicit Semantics

In last sections we reviewed metadata heterogeneities, focusing on approaches for harmonizing and encoding semantically heterogeneous information in digital libraries. However, it is not easy to apply them in the setting of P2P networks. First, the highly centralized model for creating a *union catalog* was feasible in large digital libraries, such as Library of Congress (LC), because "LC controlled the funding and could establish record creation guidelines before digitization occurred, therefore providing for a high level of interoperability with records among different institutions" [160]. However, in P2P networks many small or moderate digital libraries exist and it would be too expensive for them to create union catalog respectively. Second, one of the most popular adapted approaches — *crosswalks*, is applicable when the number of metadata elements is relatively small since mapping among different metadata schemas is less laborious. However, when more complicated metadata get involved, various difficulties will arise, as the problem of different degree of equivalency in crosswalks indicated by Zeng [135]. In addition, extensive efforts from domain experts are often needed to specify various relations between metadata elements, while automatic mapping methods are still under construction. Finally, *application profile* and registry share a common feature that they both allow particular systems to customize or adapt elements from 'core' or 'standard' schemas. Particularly, application profiles providing a dynamic and flexible mechanism to accommodate individual needs in annotating records, which is in line with the dynamic feature of P2P networks. However, formal model of application profile is still under development. Other weaknesses, as remarked by Baker[161] are: 1) information declared (redundantly) in a Metadata Vocabulary is not included; 2) it only includes information which is particular to the application profile; 3) it lacks formal schema language used to support resolution of cross-references and merging of data.

The basic problems of the approaches mentioned above all originated from the lack of an explicit model of information semantics. Recently,

it has been widely recognized that a partial explication of information semantics is required in connection with the World Wide Web. Fensel identifies a three level solution to the problem of developing intelligent applications on the web[162]:

- *Information Extraction*: In order to provide access to information resources, information extraction techniques have to be applied providing wrapping technology for a uniform access to information.

- *Processable Semantics*: Formal languages have to be developed that are able to capture information structures as well as meta-information about the nature of information and the conceptual structure underlying an information source.

- *Ontologies*: The information sources have to be enriched with semantic information using the languages mentioned in step two. This semantic information has to be based on a vocabulary that reflects a consensual and formal specification of the conceptualization of the domain— an ontology.

Information Extraction corresponds directly to the approaches using natural language processing techniques for accessing and retrieving information. Despite the progress in natural language processing, there are still many limitations due to the lack of explicit semantic information[163]. *Processable Semantics* is in connection with the syntactic and structural approaches, eg. using XML and RDF to annotate information sources. The last level of applying *ontologies* is to enrich information sources with additional semantic information. The use of ontologies has already been implemented in recent approaches for information searching in terms of meta-annotations and terms definitions[164, 165, 166]. Note that the use of explicit semantics is not contradictory to the other two approaches mentioned above, but rather an additive and powerful technique to improve or enable the other approaches. Furthermore, we think that searching in large-scale distributed environment, such as P2P networks, requires explicit semantic models.

In the following chapters, we will revisit the challenges in digital library applications in a higher level and justify the feasibility of applying Semantic Web technologies for achieving explicit semantics information sources.

5.4 Chapter Achievement

- Described the categories of metadata heterogeneities, namely syntactic, structural (schematic) and semantic heterogeneities.
- Emphasized on current approaches to semantic interoperation, such as crosswalk and application profiles, etc. and introduce typical metadata encoding methods in digital library applications.
- Pointed out that the basic problems of these approaches originate from the lack of an explicit model of information semantics. Explicit semantics is intensively needed in realizing semantic searching in digital library applications.

Chapter 6

The Semantic Web and Digital Libraries

6.1 Introduction

Evolving from traditional libraries, digital libraries concentrate on making information sources available to a wider audience. For example, scanning papers and books and preserving them. However, digital libraries only take limited advantage of the benefits modern computing technologies offer [167]. To overcome this bottleneck, research and development for digital libraries have been conducted in aspects of processing, dissemination, storage, search and analysis of all types of digital information.

As to be mentioned in Chapter 3, the P2P overlay network, as well as the grid, allows flexible, secure and coordinated resource sharing among *dynamic* digital libraries (e.g., libraries/peers can join and leave freely). However, in such a dynamic environment, often individual library uses its own specific data format unless there is a globally shared one. Thus, it is hard to see how they can *interoperate* in a meaningful manner. To address such an challenge, the Semantic Web is invented to "combine information from multiple heterogeneous sources, such as published RDF sources, personal web pages, and data bases in order to provide an integrated view of this multidimensional space" [168].

This chapter is to revisit the challenges in digital libraries, introduce the Semantic Web technologies concentrating on processing 'semantics' and justify why they are important in addressing semantic interoperation problems. Finally, Description Logics will be introduced, with a special

intention for paving road that logic-based reasoning is useful in explicating complex relations.

6.2 Digital Libraries Challenges - Revisited

Over the last decades, there has been considerable research activity in the field of digital libraries and many research challenges have turned up. Some of the key research challenges for digital libraries as sought by the US National Science Foundation are[169]:

- *Interoperability:* The ability of digital libraries to exchange and share documents, queries and services. The term also encompasses the ability to generate a single view of different libraries without forfeiting independence.
- *Description of objects and repositories:* The need to establish common schema for the naming of digital objects so as to facilitate search and retrieval from disparate distributed sources.
- *Collection management and organization:* The ability to store, index and retrieve non-textual and multimedia content.
- *User interfaces and human-computer interaction:* How information is displayed and visualized, and how the user navigates large information collections.

Obviously, assigning 'semantics' to digital objects and making them interoperable are the fundamental challenges. In current digital library applications, nevertheless, the capability of semantic search is rather limited. Consider two widely used searching strategies in digital libraries as follows:

- Using common controlled vocabularies in subject identifiers to facilitate search;
- Using a set of common metadata to describe specific information, which is further used to facilitate searching on specific fields.

By publishing controlled vocabularies in one place, which can then be accessed by all users across the Web, then library catalogues can use the same web-accessible vocabularies to catalogue their publications, marking

them using the most relevant terms from the most relevant thesauri for the domain of interest. Then search engines can use the same vocabularies to control and refine their search to ensure that the most relevant items of information are returned to the user. In contrast, various metadata can be used to describe the meaning of digital objects, using pre-defined metadata terms, such as creator, title, date, publisher, etc. It can facilitate search engines to focus on specific metadata fields without having to go through the entire context.

However, both approaches suffer from some problems in searching across similar or relevant fields described by different metadata or subject terms:

- In digital libraries, the creation and maintenance (i.e. description of objects and repositories) of standardized metadata and controlled vocabularies are usually costly activities. Often any modification on them would take a long procedure before they can be adapted in specific applications. For example, specific extensions on these standards are generally for individual usage purpose and are difficult to be harmonized with other applications if no prior agreement is reached. Even if such agreement can be reached, it is often conducted *manually* by metadata experts, e.g. in a simple mapping table.

- Annotations on digital objects by using conventional metadata generally convey no 'meanings'. In fact, both of 'flat' metadata schemas, e.g. Dublin Core and hierarchical schemas, e.g. MODS [170], are a bunch of standardized (ie. pre-defined) vocabularies which are used to facilitate 'structured' search (i.e. metadata search, c.f.Section 2.5.2) instead of 'semantic' search. With semantic search capability, we can not just support structure search, but also deduce implicit information from annotated records.

- Deep semantic interoperability, as identified as the "grand challenge" of digital libraries [169] is limited. Clearly, addressing the semantic heterogeneity has something to offer in response to all 'interoperability' challenges. This was seen as the ability to access, consistently and coherently, similar classes of digital objects and services, distributed across heterogeneous repositories with mediating 'agent' to compensate for site-by-site variations.

These problems are also the origins for the challenges aforementioned. Thus, new technologies are required to facilitate semantic enrichment for digital objects and further be able to harmony semantic dissimilarities. The Semantic Web technologies can then be applied for this purpose.

6.3 The Semantic Web

6.3.1 Brief History of the Semantic Web

Prior to the inception of the Semantic Web, the World Wide Web (WWW) provides interoperability at various levels. For examples, the TCP/IP protocol furnishes a robust way to transport data from node to node; the HTTP and HTML offer a standard way to retrieve and represent hyperlinked textual documents. However, due to the huge volume of online documents and the insufficient representation of knowledge contained in them, machines are found crippled in processing 'semantics' in documents. By using HTML, one can create and present a page that lists books she is interested in, such as "BrokenBack Mountain", but she may not be able to use HTML to unambiguously assert, for example, that Book01 is named "BrokenBack Mountain" and authored by "Annie Proulx". In addition, there is also no way to express that "BrokenBack Mountain" is a mytitle or "Annie Proulx " is a mycreator.

To overcome such shortcomings, ontologies recently have become a topic of interest in computer science. Equipped with ontologies, the Semantic Web is able to make web resources - not just web pages, but also a wide range of web accessible data and services - more understandable to machines. In other words, machines would not just be able to display data, but rather be able to use it for automation, integration and reuse across various applications.

6.3.2 Ontology - The Key Enabler for the Semantic Web

Originated first in philosophy, Ontology is in computer science claimed to be an 'explicit specification of a conceptualization that facilitates knowledge sharing and reuse'[171], 'content theories about the sorts of objects, properties of objects, and relations between objects that are possible in a specified domain of knowledge' [172], and 'an entity-relationship schema with subsumption relations between concepts'[165]. From a knowledge engineering perspective, ontologies are constructed using specialization-

generalization relationships to form their *taxonomies* and using other semantic *relationships* (e.g. part-whole, derivationally related) to extract the meaning of concepts and factual knowledge of a domain.

More concretely, an ontology can be viewed as a generalization of a taxonomy as illustrated in Figure 6.1.

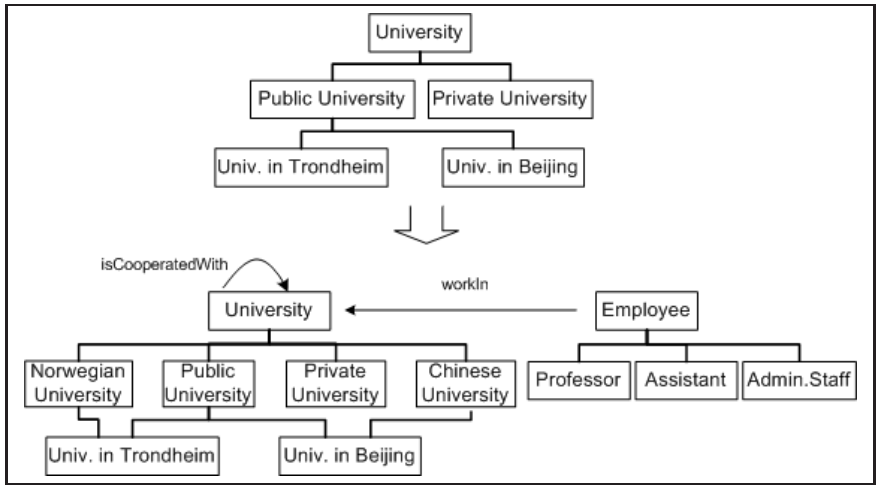

Figure 6.1: Example: University Taxonomy and Ontology

At the top of Figure 6.1 is a simple taxonomy of Universities, divided into public and private, and with the former subdivided into those in Trondheim and Beijing.

Below is a much richer structure. the simple taxonomy of Universities has been expanded to include the class of Norwegian University and Chinese University, such that Univ. in Trondheim and Univ. in Beijing are each a subclass of two superclasses. The existence of more than one superclass of a given class is a feature not normally found in conventional taxonomies [173].

In addition, one university can cooperate with other universities. And the property workIn has as its subject an Employee and its object a University. Note that properties are inherited by subclasses. Thus, since Univ. in Trondheim has two superclasses, it can inherit properties from both.

In the context of the Semantic Web, ontologies provide a shared understanding of a domain of interest to help automated processes (i.e. "intelligent agents") to access information, typically represented in a machine-processable language. In addition, ontologies are expected to be used to

provide structured vocabularies that explicate the relationships between different terms, allowing intelligent agent s to interpret their meaning flexibly yet unambiguously [174]. Moreover, terms whose meanings are defined in ontologies can be used in semantic markup that describes the content and functionality of Web-accessible resources [175].

In nutshell, developing ontologies help to represent knowledge in a machine processable way, enabling the *relationships* to be described; they express a shared view on a domain of interest.

6.3.3 The Semantic Web Languages

Together with the development of the Semantic Web, a set of standards, or in other words, a set of Semantic Web languages have emerged for different 'semantic' requirements on various applications. Before we introduce how to implement Semantic Web technologies in digital libraries, it is necessary to look through these standardized languages

XML

The basic building block, XML [155] provides a formal syntax for describing documents in a richly structured manner, as different from HTML which trades description power for ease of use. A typical XML element, to describe an attribute of "BrokenBack Mountain", might be $<$ *mycreator* $>$ *AnnieProulx* $<$ */mycreator* $>$. Although it is intuitively obvious, in practice much effort has to be put into developing agreed terms such as mycreator in our example. Nowadays XML has been widely used as an interchange language over a range of business applications. However, XML does not impose semantic constraints on the meanings of documents [1]. Thus we have no way of describing anything about "BrokenBack Mountain", e.g. that it is yet the name of an Oscar movie and the director is "Ang Lee". To overcome this limitation, Resource Description Framework (RDF) was designed.

RDF

RDF is a simple data model for referring to resources and how they are related, with an intention to exchange information between applications without loss of meaning [153].

[1] As a 'datatyping' language for restricting the structure of XML documents, XML Schema provides limited semantic description capabilities.

An RDF statement (or RDF triple) is of the form:

[Subject Property Object]

Herein, a Subject (i.e. an RDF resource) is usually named by a URI (c.f. Chapter 1) - this includes all the Web's pages, individual elements of an XML document, or even a 'blank node' [176]. A Property generally has a name and can be used as "attribute", e.g. *price* or *title*. Practically, many elements in *flat* metadata schemas, such as Dublin Core[9], are used as property; but some elements in *non-flat* schemas like MODS, ABC, etc are also used to annotate resource (subject). RDF annotates Web resources in terms of named Property. Values of named properties (i.e. Object) can be URIs, Web resources or literals (i.e.data values, such as integers and strings Note that not only Subject, but also Property and Object can be Resources, which is argued as an effective way to do lookups based on other people's metadata [157].

To represent RDF statement in a machine-processable way, RDF uses XML syntax, referred to as RDF/XML [153] or Notation 3 (or N3) syntax of RDF. For verbosity reason, we use in this thesis N3, where each RDF statement is of the form in Figure 6.2.

```
@prefix rdf: < http://www.w3.org/1999/02/22 − rdf − syntax − ns# >
@prefix ex: < http://example.org# >
@prefix bm: < http://www.bk.com >

bm:Book01 ex:mytitle "BrokenBack Mountain" ;
          ex:mycreator "Annie Proulx" ;
          ex:mypublisher _:b1 .
_:b1 bm:name "Scribner"
```

Figure 6.2: RDF Data Model

In Figure 6.2, "@prefix" introduces shorthand identification of XML namespaces and a semicolon ";" indicates another property of the same subject. In the statements, the resource is bm:Book01, which has two properties ex:mytitle and ex:mypublisher. Note that _:b1 is a blank node identifier.

In brief, RDF does have some advantages over other alternatives such as XML. It is a generic open standard whereas many alternatives are either proprietary or specific to a particular domain. It is the first time to make possible describing statements via standardized data model (together with serialization syntax) whereas direct use of XML focuses just on the document syntax. By breaking down information into small independent

units (triples) and using global identifiers for all objects/properties/types (URIs) it becomes possible to integrate information from several sources by simply concatenating the sets of the triples and following the new relations. The data model is sufficiently simple and makes sufficiently few assumptions that it be used to express both structured and semi-structured data making integration across heterogeneous sources more straightforward.

RDFS

Often one may need, for example, to specify that "Annie Proulx" is an instance of the class "Person". Moreover, one may also want to define relationships, e.g., teachIn as having a specific domain (the instances in the class "Person") and range (the instances in the class "University"). Although RDF provides a standard *syntax* to create, exchange and use statements in the Semantic Web, it does not enable us to *describe* semantics. RDF Schema (RDFS)[146] is thus designated to handle such problems.

In RDFS, predefined Web resources rdfs:Class,rdfs:Resource,rdfs:Property can be used to define classes (i.e. concepts), resources and properties (i.e. roles) respectively. In addition, a set of meta-properties are also used to represent background assumption in ontologies, namely, rdf:type, rdfs:subClassOf, rdfs:subPropertyOf, rdfs:domain and rdfs:range.

RDFS statements are typically RDF triples. Figure 6.3 illustrates a fragment of the ontology shown in Figure 6.1.

```
@prefix rdf: < http://www.w3.org/1999/02/22-rdf-syntax-ns# >
@prefix rdfs: < http://www.w3.org/2000/01/rdf-schema# >
@prefix univ: < http://www.univ.edu >

univ:University rdf:type rdfs:Class .
univ:Employee rdf:type rdfs:Class .
univ:PublicUniversity rdf:type ; rdfs:subClassOf univ:University .
univ:workIn rdf:type rdfs:Property ;
        rdfs:domain univ:Employee ; rdfs:range univ:University .
```

Figure 6.3: An RDFS ontology

However, RDFS also suffers some limitations. For examples, insufficient expressiveness which disallows use to express 'a University is different from a Employee' because *negation* is not supported [177]; few restrictions on its syntax which can easily leads to confusions [178]. In brief, RDFS

is a limited ontology language that supports only class and property hierarchies, as well as domain and range constraints for properties. Hence to describe advanced more complicated meanings it is crucial to have a more advanced ontology language having [174].

OWL

Web Ontlogy Language (OWL) [154] facilitates greater machine interpretability of Web content than that supported by RDF and RDFS (also referred to RDF(S)) . OWL can declare classes and organize these classes in a subsumption ('subclass') hierarchy, as is basic capabilities of RDFS. Going beyond the capability of RDFS, OWL can specify classes as logical combinations (intersections, unions or complements) of other classes, or as enumerations of specified objects. As to properties, in addition to declaring properties as in RDFS by applying subProperty, domain and range, OWL can also state that a property is transitive, symmetric, functional, or is the inverse of another property.

However, the major extension over RDFS is that OWL support new axioms (constraints) along with formal semantics. For examples, OWL can define all values for a property of instances of a class must belong to another class (or datatype); at least one value must come from a certain class (or datatype); and there must be at least or at most a certain number of distinct values.

The design of OWL is also subject to a variety of influences. As remarked in [174], these included influences from established formalisms and knowledge representation paradigms, influences from existing ontology languages and influences from existing Semantic Web languages (e.g. RDF(S)). We will come to this point in Chapter 7.

6.4 Description Logics and Ontologies

Many approaches to information integration have proved the belief that it is best to be presented at a conceptual and higher level. Description Logics [2], first appeared in [179], are a family of object-centered knowledge representation languages that can be used to represent the knowledge of an application domain in a *structured* and formally well-understood way [180, 181, 19, 182, 183, 184, 185, 186]. A main point is that DLs are

[2] http://dl.kr.org/dl

considered as to be attribute logics in knowledge based applications as they are a good compromise between expressive power and computational complexity [187]. Therefore, DLs can be used in many ontology integration applications to describe data semantics and support inferencing.

Here we take an example of describing an extended class $ReferentialExpression$ in FRBR [46] as follows[3]:

$ReferentialExpression ::=$

$Expression \sqcap \neg AutonomousExpression \sqcap (\geqslant 1$ isReferentiallyRelatedToExpression.Expression)

It means "A *ReferentialExpression* is something that, amongst other things, is a *Expression* but is not a $AutonomousExpression$ and has *at least one* referentially related Expressions".

In fact, DLs have been widely used in many ontology languages, especially those are described in DLs, such as OIL, DAML+OIL and OWL [188], which indeed take major roles in building the Semantic Web.

We present next the formal specification of DLs.

6.4.1 Formal Syntax and Semantic of Description Logics

The basic notion in description logics is that they regard a world of entities that can be grouped into classes, called *concepts* and that can be related to each other by binary relationships, called *roles* [5]. A typical description logic contains several elementary notions: atomic concepts A and atomic roles P; universal concept $\top$ and bottom concept $\bot$. More complex concepts and roles from simple one, by applying additional *constructors* in description logics, such as, the Boolean constructors *conjunction* ($\sqcap$), and *negation* ($\neg$), as well as the *existential restriction* constructor ($\exists\ R.C$), the *value restriction* ($\forall\ R.C$), and the *number restrictions* constructor ($\geq nR$).

These constructors in DLs provide different expressive power. In general, DAML+OIL and OWL support $\mathcal{SHIQ}(\mathcal{D})$ [189] of description logic. In Table 6.1), we summarize the syntax and corresponding semantics of DLs constructors.

In addition to this description formalism, DLs are usually equipped with a *terminological* (TBox) and *assertional* (ABox) formalism. Generally, terminological axioms are used to introduce names (abbreviations) for complex descriptions, while the assertional formalism are used to state properties of individuals [5]. A TBox, is generally a collection of axioms of the form $\alpha \sqsubseteq \beta$ or $\alpha \equiv \beta$. For every atomic concept A, there is at most one axiom in TBox whose left-hand side is A. In the ABox, one describes

[3]For simplicity, we describe them in Prolog syntax

Table 6.1: Syntax and Semantics of Description Logic Constructors

	Construct Name	Syntax	Semantics
$\mathcal{S}$	primitive concept	A	A^I
	primitive role	P	P^I
	universal concept	$\top$	Δ^I
	bottom concept	$\bot$	$\emptyset$
	conjunction	$C_1 \sqcap \ldots \sqcap C_n$	$C_1^I \cap \ldots \cap C_n^I$
	disjunction	$C_1 \sqcup \ldots \sqcup C_n$	$C_1^I \cup \ldots \cup C_n^I$
	negation	$\neg C$	$\Delta^I \setminus C^I$
	subsumption	$C_1 \sqsubseteq C_2$	$C_1^I \subseteq C_2^I$
	equivalence	$C_1 \equiv C_2$	$C_1^I \equiv C_2^I$
	value restriction	$\forall P.C$	$\{x \in \Delta^I \mid P^I(x) \in C^I\}$
	existential restriction	$\exists P.C$	$\{x \in \Delta^I \mid P^I(x) \cap C^I \neq \emptyset\}$
$\mathcal{H}$	role subsumption	$P_1 \sqsubseteq P_2$	$P_1^I \subseteq P_2^I$
	role equivalence	$P_1 \equiv P_2$	$P_1^I \equiv P_2^I$
$\mathcal{O}$	nominal I	$\{o\}$	$\{o\}^I \subseteq \Delta^I, \mid \{o\} \mid = 1$
$\mathcal{I}$	role inverse	P^-	$\{(y,x) \mid (x,y) \in P^I\}$
$\mathcal{N}$	number restrictions (cardinality)	$\leqslant$ nP	$\{x \in \Delta^I \mid \mid P^I(x) \mid \geq n\}$
		$\geqslant$ nP	$\{x \in \Delta^I \mid \mid P^I(x) \mid \leq n\}$
		$=$ nP	$\{x \in \Delta^I \mid \mid P^I(x) \mid = n\}$
$\mathcal{Q}$	qualified number restrictions	$\leqslant$ nP.C	$\{x \in \Delta^I \mid \mid P^I(x) \cap C^I \mid \geq n\}$
		$\geqslant$ nP.C	$\{x \in \Delta^I \mid \mid P^I(x) \cap C^I \mid \leq n\}$
		$=$ nP.C	$\{x \in \Delta^I \mid \mid P^I(x) \cap C^I \mid = n\}$
$\mathcal{D}$	Datatype	D	$D^I(x) \subseteq \Delta_D^I$
	datatypeProperty	T	$T^I \subseteq \Delta^I \times \Delta_D^I$
	value restriction on data range	$\forall T.C$	$\{x \in \Delta^I \mid T^I(x) \in C^I\}$
	existential restriction on data range	$\exists T.d$	$\{x \in \Delta^I \mid T^I(x) \cap C^I \neq \emptyset\}$

a specific state of affairs of an application domain in terms of concepts and roles and some of them may be defined by names of the TBox. In the ABox, by denoting individual names as a, b, c. Using concepts C and roles R, one can make assertions of the following two kinds in an ABox: $C(a)$; $R(b; c)$

A description logic system not only stores terminologies and assertions, but also offers services that reason about them. We are to extend the discussion in the next section.

6.4.2 Description Logics-based Reasoning

Information reasoning makes it possible to deduce new knowledge from already specified knowledge (e.g. rules) in the Semantic Web. In general, there are two different approaches. Problem-solving oriented solutions[190, 191], on one hand, are rendered via specific algorithm to proceed from a given state to a desired goal state. It is part of the large problem process that includes problem finding and problem shaping.

On the other hand, the most widely applied one is the general logic-based approach, which exploits inference engines, so as to facilitate machine understanding of information resources. Given that many ontology languages are based on description logics for the benefit of a balance between expressive power and computational complexity, description logics offer services that reason about terminologies and assertions. Figure 6.4 sketches a knowledge representation system based on DLs.

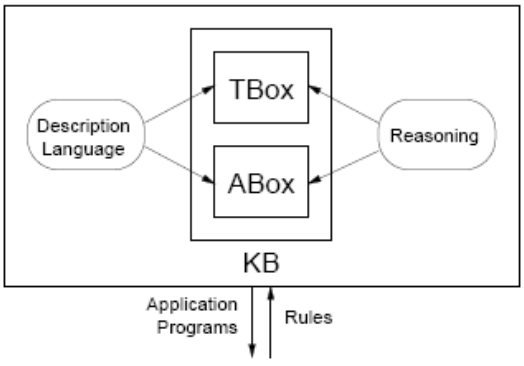

Figure 6.4: Architecture of a Knowledge Representation System based on Description Logics. From [5] (pp.50)

In description logics, typical reasoning tasks for a TBox are to de-

termine whether a description is *satisfiable* (i.e., non-contradictory), or whether one *subsumes* another one. The important tasks in an ABox are to find out whether its set of assertions is *consistent*, that is, whether such assertions entail that a particular individual is an instance of a given concept. According to [5], *satisfiability* checks of descriptions and *consistency* checks of sets of assertions are useful to determine whether a knowledge base is meaningful at all; while by the *subsumption* tests, one can organize the concepts of a terminology into a hierarchy according to their generality. In an analogous speaking, a concept description can be conceived as a query, describing a set of objects involved. Thus, the answering becomes a phase of retrieving the individuals that satisfy the query.

Various relationships have already been studied in description logics, for example, the "isA", "inverseOf" and "theSameAs" relationships. Basically, we can divide them into three major types set out as follows:

- Synonyms: When two different ontologies have the same semantics, they have a synonym relation with each other. "theSameAs" is an example;

- Hyponyms: When a term in one ontology is semantically more specialized than another term in another ontology, they have a hynonym relation. Eg. "Undergraduate" is hyponym to "student".

- Hypernym: When a term in one ontology is semantically more general than another term in another ontology, they have a hypernym relation. Eg. "Person" is hypernym to "student".

In addition, a synonym relation ($\alpha \equiv \beta$) can be further substituted by two subsumption axioms, $\alpha \sqsubseteq \beta$ or $\alpha \sqsupseteq \beta$. Indeed, most of these relations are hierarchical or similarity based. However, such relations (i.e., subsumption) are not powerful enough for our task of semantic searching across heterogeneous domains, such as a bibliographic publication repository and a personal paper collection where may need specify many-to-one, one-to-many, or other complex relations. Thus, it is necessary to conduct a further investigation on possible relationships between different metadata terms, or generally, ontology concepts, so as to facilitate new solutions to bypass these problems.

6.5 Inferential Rule based Ontology Translation

Sometimes the expressive power of description logic based ontology languages is too limited to fulfill advanced requirements in specifying complex relations. To strengthen the expressive capability, reasoning turn out to be a feasible mechanism, deducing implicit relations or knowledge out of predefined rules and explicit facts. Such an approach can be deployed in the Semantic Web stack [6] (c.f. Figure 6.5) where upper 'rules' are used to extend the capability of 'ontology' below. Since the instantiation of the OWL, attention has turned to the rule layer [192].

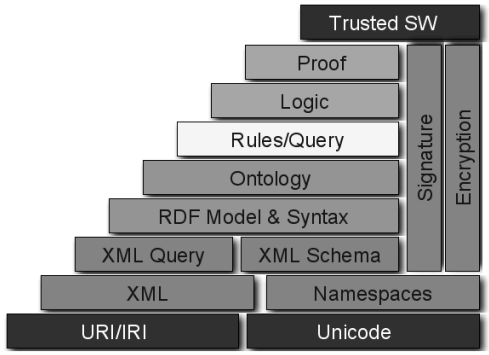

Figure 6.5: Semantic Web Track. From [6]

Different kinds of rules might provide us various and flexible methods to resolve semantic gaps among ontologies, as remarked in [193]. For examples, standard-rules, for chaining ontologies properties, such as the transfer of properties from parts to wholes; bridging-rules for reasoning across domain; mapping rules between Web ontologies for data integration; querying-rules for expressing complex queries upon the Web; meta-rules for facilitating ontology engineering (eg. acquisition, validation, maintenance). Further discussion on combining ontologies with inferencing rules is in Section 8.5.

6.6 The Importance of Applying Semantic Web in Digital Libraries

Digital libraries generally contain a large volume of digital documents, focusing on making their information resources to a wider audience. Within a pool of heterogeneous and distributed information resources, there is no excuse for letting users take site-by-site searching. Thus, considerable effort is required in creating meaningful metadata, organizing and annotating digital documents, and making them accessible.

Semantic Web technologies introduced previously provide a set of standards and languages to facilitate the description of objects and repositories, such as help establishing common schemas in form of ontologies. Generally, disadvantages in applying semantic technologies come from extra cost in computing relationships between concepts/properties. In addition, they focus more on local and static situations, rather than a distributed and dynamic environment. However, these disadvantages turn out to be less important because of more and more powerful processing capability. Furthermore, by combining with other decentralized infrastructures, eg. peer-to-peer system, the full potential of the Semantic Web can be exploited. In other word, Semantic Web technologies can be used to search distributed information by using peer-to-peer systems as supportive platform.

In fact, Semantic Web technologies has been used in digital libraries in aspects of user interfaces, human-computer interaction, user profiling, personalization and user interactions [168]. However, the major capability of Semantic Web technologies is to enable information sharing in a semantic way, thus it is significant to extend them to address the *interoperability* issue - the "grand challenge" in digital libraries (c.f. Section 6.2).

In the following chapters, we are to describe the process of enriching metadata with semantic meanings for facilitate semantic searching across distributed and autonomous digital libraries.

6.7 Chapter Achievement

- Justified the heterogeneity problem, especially the semantic interoperability problem is the most challenging one in digital library applications.
- The Semantic Web languages have been introduced and relations

among them are also described. Ontologies are justified as a key enabler in the Semantic Web applications since they can provide a shared understanding of a domain of interest which is critical to semantic interoperability.

- Description Logics, the basis of several well-known ontology languages including OIL, DAML+OIL and OWL, have been introduced from a conceptual and formal level. Therefore, DLs can be used in many ontology integration applications to describe data semantics and support inferencing.

- Predefined inferencing rules can be used to elicit implicit relations between concepts/properties in ontologies which are sometimes out of the expressive power of Description Logic based ontology languages.

- Summarized that digital libraries applications can benefit from applying Semantic Web technologies, with an emphasized issue for semantic search.

Chapter 7

Semantic Enriched Metadata Management

7.1 Introduction

In conventional digital libraries, often information searching evolves into a process of keywords matching on multiple indexed 'metadata fields' (eg. music titles and authors). The results, however, frequently contain irrelevant information, but at the same time ignore information that contains similar content. Solutions based on distributed databases have the same problem — although direct and explicit facts about digital objects can be stored in database, description of the implicit relations among different types of information is limited. As pointed out by Huwe[194], users of such digital libraries are not satisfied with the quality of the depth and relevancy of information they gather. Researchers thus have to find a better way in not only interconnecting autonomous digital libraries, but also searching across implicitly related information records. Semantic Web technologies, as presented in previous chapter, turns out to be a feasible solution for achieving semantic information searching and sharing. In this chapter, we present the process of generating ontology-enriched metadata records in local sources. A general process is to be presented for creating ontological knowledge sources. In addition, different approaches on ontology interoperation, namely merging, translation and mapping, will be described and compared with a special concern on dynamic P2P computing environments.

7.2 The Role of Metadata, Context and Ontologies

In digital library applications, the goal of various metadata standards is to describe the context/meaning of digital records in a more explicit way by tagging the digital records with more 'signs'. These *metalevel* signs themselves could have further interconnections, such as being tagged with *metametalevel* signs. However, the ultimate source of meaning is the physical world and the agents who use signs (i.e. metadata) to represent entities in the world and their intentions concerning them[195]. That is, human intervention is required to appreciate these metadata unless further mechanism, e.g., interpreter, is appended.

In order to interoperate among heterogeneous but relevant metadata terms, Kashyap and Sheth [7] proposed to link the metadata terms to ontological terms, basing on an analysis on a global information system where many different and possibly heterogeneous information repositories could be integrated. Figure 7.1 illustrates the mechanism of using ontologies to expose relations between terms explicitly.

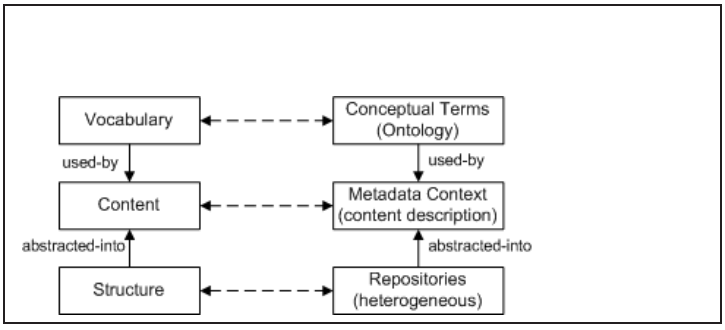

Figure 7.1: The Relations Among *Context*, *Metadata* and *Ontology*. Adapted from [7].

We bind in this thesis our concerns on metadata to that in [7]. Based on this view, we investigate how to deploy ontologies to represent relations between metadata elements in a semantic-enriched manner. That is, the issue of metadata mapping evolves into processing terminological relationships between metadata elements across ontologies. Take one simple example: suppose we have two metadata terms - University and Staff. It may be easy for human brain to tell that there is a relationship - workIn

between University and Staff. Unfortunately, current metadata based system was found incapable of describing such relationship in a standard and explicit manner. Herein, with the help of standard ontology languages, such as RDF(S), we can *explicitly* specify implicit relationship in a formal way as illustrated in Figure 7.2. According to levels of *formality* presented

```
@prefix rdf: < http://www.w3.org/1999/02/22-rdf-syntax-ns# >
@prefix ex: < http://example.org# >
@prefix univ: < http://www.university.edu >
univ:University a rdfs:Class ;
univ:Staff a rdfs:Class ;
univ:workIn a rdfs:Property ;
            rdfs:domain univ:Staff ;
            rdfs:range univ:University
```

Figure 7.2: Using RDF(S) to Represent Relations between Metadata Terms

in [196], ontology itself may have varied types as specified as follows:

- *Informal Ontology:* it is the simplest type and comprises of a set of concept names/words organized in a hierarchy.
- *Terminological Ontology:* it consists of a hierarchy of concepts defined by natural language definitions (e.g., WordNet [197]).
- *Formal Ontology:* it further includes axioms and definitions stated in a formal language such as OWL [154] and Description Logic.

In this thesis we focus more on terminological ontology and formal ontology. Approach on using axioms for knowledge inferencing will be discussed in next chapter.

7.3 Developing Ontological Knowledge Sources

While many approaches are available around the creation and management of ontologies, a thorough and systematic process is required for using ontologies in various applications. By referring to the *Knowledge Process*[198] which has been successfully applied in *On-To-Knowledge Methodology (OTKM)* project[199], we create a process (see Figure 7.3) for developing ontological knowledge sources with semantically enriched metadata.

Essentially, the process revolves around the following steps, and we will discuss them in coming sections.

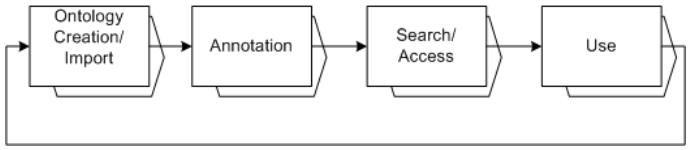

Figure 7.3: The Development Process for Ontological Knowledge Sources.

- *Ontology Creation or Import*: The specific ontology shall be created or imported from other resources so as to fit the conventions of local metadata repositories.
- *Annotation*: Implicit semantics shall be represented in an explicit way via annotation, eg. generate interlinkages among relevant records by creating relational metadata — based on available ontologies.
- *Search and Access*: This step satisfies the searches and queries for information/knowledge by average users.
- *Use*: The domain experts, or so called knowledge workers, will process returned results for further use.

7.3.1 Ontology Creation or Import

Ontology creation and import are the first phase in the enriching metadata with semantics. Various methodologies exist to conduct the theoretical approach and different types of ontologies (ie. informal, terminological, formal ontologies) in different granularities (ie. from specific domains to interdisciplinary domains) can be created. The general ontology creation process is illustrated in Figure 7.4:

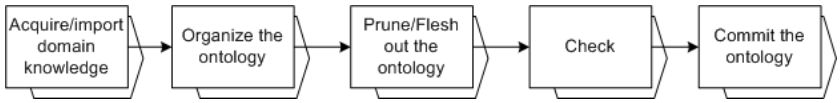

Figure 7.4: The General Ontology Creation Process.

In this process, domain experts (ie. knowledge engineers) must identify each concept that is comprised of a terminology, attributes, and relationships among terminologies. Numerous concepts can then be structured as taxonomy and become domain ontology.

Building ontologies often involves many ontologies from external sources as well as existing and newly developed in-house ontologies since creating a global ontology has been justified infeasible[200]. Often these available ontologies can be directly imported into building new ontologies. However, re-organization, pruning and fleshing concepts, relations and individuals are still necessary to satisfy changed scenarios. The subsequent works are checking inconsistencies among ontology elements, verifying the final version of ontology and publishing it within its intended deployment system.

A critical task in this process is to efficiently elicit concepts from knowledge and create ontology structure. Herein, Formal Concept Analysis (FCA) method [201] has been justified as a feasible approach for facilitating knowledge acquisition. FCA is a mathematical approach that analyzes relationships among components and calculates their dependency. First, this method computes and draws a concept lattice to express the components of the knowledge domain, including objects, attributes, and their relations. Second, mathematic algorithms are used to calculate dependency rates and derive implications regarding their relations. Finally, a terminology hierarchy is proposed and a concepts hierarchy can be considered.

The research and projects in the Semantic Web have brought us a plethora of ontology editors, each having its own specialities and functionalities. These editing tools can help in accomplishing most aspects of ontology creation, such as map and link between them, compare them, reconcile and validate them, merge them, and convert them into other forms. Despite the immaturity of this field, a bunch of ontology editors are identified — more than 50 overall[202]. Among them some prominent ontology editors are: Protégé[1], OntoEdit[2] and KAON[3].

Protégé is probably the most well-known free and open ontology editor. The Protégé platform supports two main ways of modeling ontologies via the *Protégé-Frames* and *Protégé-OWL editors*. At its core, Protégé implements a rich set of knowledge-modeling structures and actions that support the creation, visualization, and manipulation of ontologies in various representation formats. Protégé can be customized to provide domain-friendly support for creating knowledge models and entering data. Protégé ontologies can be exported into a variety of formats including RDF(S), OWL, and XML Schema.

[1]http://protege.stanford.edu
[2]http://www.ontoprise.com
[3]http:/kaon.semanticweb.org

OntoEdit is a commercial ontology editor. Similar to Protégé, it is based on a flexible plug-in framework. OntoEdit has strong inferencing capabilities - with interfaces to several information engines.

KAON is not just an open-source ontology editor, but rather an ontology management infrastructure targeted for business applications. It includes a comprehensive tool suite allowing ontology creation and management. In addition, it provides a framework for building ontology-based applications. An important focus of KAON is efficient reasoning with ontologies.

7.3.2 Enriching Metadata Records with Semantics

As Marshall pointed out, annotations may take many forms and can be conducted both through automatic and human means[203]. A rough classification of annotation types (cf.[204]) is as below:

- *Textual Annotation*: Annotation of this kind adds extra notes are to metadata records. It is a conventional method which has been applied for a long time. For example, in the SWISS-PROT Protein knowledgebase[205], commentary information, such as functions, structure, domains and so on, are also added to specific protein sequence information. To make it distinct from *Semantic Annotation*, we assume that textual annotation is not accessible to machine-processing.

- *Link Annotation*: It extends the textual annotation notion, where the content of the annotation is given, not by some text, but by a *link* destination and possibly associated behavior. This kind of annotation is also targeted as human readers.

- *Semantic Annotation*: Semantic annotation is to tag ontology class instance data and map it into ontology classes. Annotation of this kind is targeted for machine-processing — this does bring with the requirement that implicit relationships be explicitly represented. The idea of semantic annotation has been pursued in many projects, such as Ontobroker[206], SHOE[207] and recently CCOHSE[208].

What we want to emphasize is that *fully* automatic semantic annotations remains an unsolved problem in the process of annotation because it is not yet possible to automatically identify and classify all entities in

source documents with *complete* accuracy[209]. Instead, semi-automatic approaches are often applied, relying on human intervention at some point in the annotation process[210].

Semantic annotation serves as a significant role in enabling semantic search. That is, programs running on the machines (ie. Agents) need metadata that describes the content of digital objects to perform searching over such resources. Furthermore, information reasoning can be rendered over rich semantic metadata, enhancing the possibility of retrieving semantically related records.

In digital libraries - the digital face of traditional libraries, digital records are simply annotated according to *prior agreed* metadata schemas. For example, the Dublin Core Metadata Element Set [9] provides 15 *core* properties, such as title, subject and date, etc. with descriptive semantic definitions. One can use these information properties in RDF or even META tags in HTML documents. For example, statements in Figure 6.2 can be rewritten as shown in Figure 7.5.

```
@prefix rdf: < http://www.w3.org/1999/02/22 − rdf − syntax − ns# >
@prefix dc: < http://purl.org/dc/elements/1.1/ >
@prefix bm: < http://www.bk.com >

bm:Book01 dc:title “Brokenback Mountain” ;
          dc:creator “Annie Proulx” ;
          dc:publisher _:b1 .
_:b1 bm:name “Scribner”
```

Figure 7.5: RDF Data Model

In Figure 7.5, dc:title, dc:creator and dc:publisher (cf. Figure 7.5) substitue ex:mytitle, ex:mycreator and ex:mypublisher(cf. Figure 6.2) respectively. Thereby, Dublin Core compatible software agents running on distributed computers can then understand that the title of the Web resource is “Brokenback Mountain”, and the creator is “Annie Proulx”. This is beyond the promises of the statements shown in Figure 6.2 because users can simply map their elements to corresponding Dublin Core elements which are shared in a wider landscape. The limitation, at the same time, is that we can not expect a rich set of pre-defined terms due to the cost.

To cope with the limitation, ontology-based approach is used to specify not only the meaning of metadata terms, but also the relationships between them. The feature ‘representation of a shared conceptualization’ enables concepts, their relations and even constraints in a domain to be communicated between people and heterogeneous and distributed

systems. In contrast, the ontology approach is more flexible than the *prior agreements* approach because users have more freedom in customizing concepts, relations and constraints in ontologies. Typically, users can specify the meaning of digital objects (through annotations) by asserting resources as instances of certain concepts and relate a resource to another resource via some properties defined in ontologies.

7.3.3 Semantic Information Search

Semantic search in this thesis is related to retrieving more relevant results from repositories scattered across distributed digital libraries, rather than searching in a non-straightforward and uninterpretable manner. For example, a search query like "semantic day seminar" does not denote any concept and the system just tries to find metadata records containing all these words. However, if one searches "digital library" which denotes a subject in computing classification system, the system may return *augmented* results covering "information search and retrieval" which is relevant to "digital library". More specifically, domain ontologies are used to expand search arguments so as to increase results relevance. Query reformulation is done by transforming original query and using concepts of the domain ontology.

7.3.4 Semantic Information Usage

The Semantic Web is an open environment in which applications do not commit on the use of a unique ontology and can always find improvement in ontologies to better describe/annotate specific content. From searching results returned in previous step, domain experts could test whether concepts are consistent and integrated and whether one concept has intended meaning or other derived consequences. As a consequence, further implicit information or rules may be found and used to flesh out available ontologies. It thus leads to an iterative approach illustrated in Figure 7.3.

7.4 Interoperating Semantically Heterogeneous Sources

7.4.1 Current Approaches

The previous section presents the process for creating semantically enriched metadata records in source repositories, while in this section approaches to interoperating with semantically heterogeneous sources will be discussed.

Different approaches[211, 212, 196, 213] have been conducted recently in aspect of sharing heterogeneous information. Among them the most discussed ones are *merging*, *translation* and *mapping*, as illustrated in Figure 7.6.

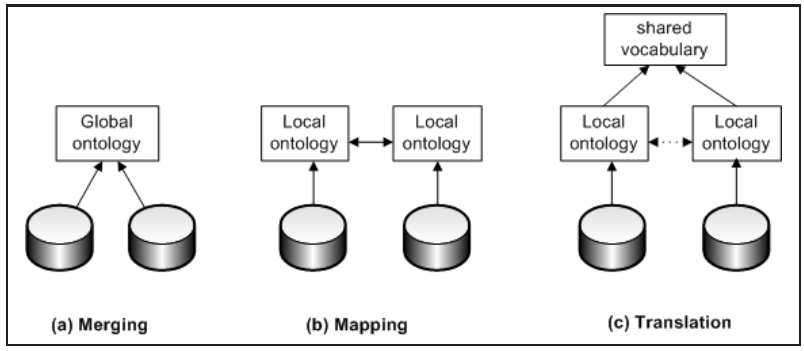

Figure 7.6: The Most Often Applied Methods in Inter-relating Ontologies

Merging

Ontology merging is achieved by merging several source ontologies into a *single* target one that unifies all of them[214]. In general case, the source ontologies would be removed and only the target (merged) ontology remains. In other words, queries rendered to all participating sources are formatted according to the merged ontology. A special case is that the source ontologies are still in use, but along with mappings to the merged ontology.

The advantage of this approach is thus that the merged ontology contains all information that is needed for interoperation. That is, the merged ontology virtually substitutes source ontologies.

Often ontology merging is conducted in a bottom-up manner although knowledge experts often build ontologies from the top down with grand conceptions[215]. The weakness is that great efforts are needed to come up with a merged ontology that correctly resembles all participating ontologies[216]. Obviously, there are considerable overheads in merging a large number of ontologies, or in situation that ontologies themselves are large (ie. many concepts and relations). Furthermore, inconsistency would happen in the resulting ontology[217]. To resolve such inconsistencies, some definitions must be changed, or some of the types must be relabeled as well.

Mapping

In constrast to ontology merging, the process of ontology mapping explicates relations between two individual ontologies at conceptual level and *mutual* interpretations will be established from both sides of systems. I [218]. That is, instances of the source ontology need to be transformed into the form of the target ontology entities according to those semantic relations. To achieve such a function, linguistic, statistical, structural and logical methods can be used (cf. summary from Harmelen[219]).

- *Linguistic Methods*: They try to exploit the linguistic labels attached to the concepts in source and target ontologies in order to discover potential matches. This can be as simple as basic stemming techniques or calculating Hamming distances, or can use specialized domain knowledge. An example of this would be that the difference between *Diabetes Melitus type I* and *Diabetes Melitus type II* should be removed by a standard stemming algorithm.

- *Statistical Methods*: They typically use instance data to determine correspondences between concepts: if there is a significant statistical correlation between the instances of a source-concept and a target-concept, there is reason to belief that these concepts are strongly related (by either a subsumption relation, or perhaps even an equivalence relation). These approaches of course rely on the availability of a sufficiently large corpus of instances that are classified in both the source and the target ontology.

- *Structural Methods*: They exploit the graph-structure of the source and target ontologies, and try to determine similarities between

these structures, often in coordination with some of the other methods: if a source- and target-concept have similar linguistic labels, then dissimilarity of their graph-neighborhoods can be used to detect homonym problems where purely linguistic methods would falsely declare a potential mapping.

- *Logical Methods*: They are most specific to mapping ontologies. Instead of simply mapping record-fields or database-schemata, additional rules could be established indicating relations between source- and target-concept or attributes.

Today, mappings are still largely conducted by hand, in a labor intensive and error-prone process. As a consequence, semantic interoperation/integration issues have become a key bottleneck in the deployment of a wide variety of information and knowledge management applications. The tension of this bottleneck has motivated numerous research activities [220, 221, 222] in aspects of describing mappings, processing mapping and generating them automatically/semi-automatically.

Translation

Ontology translation tries to combine both of the advantages of ontology mapping and merging. Generally, ontology translation will use a set of controlled vocabularies or third party ontology as background knowledge when mapping between a source and a target ontology. Two major methods are *shared vocabulary based approach* and *direct source-to-target approach*. The idea of the former approach is to define a *shared/controlled vocabulary set* which be adopted by participating sources. That is, mappings only have to be established with the shared vocabulary and relations with other sources are acquired by terminological reasoning. Ontolingua [93] is a typical example of this approach. The advantages of this approach are its scalability since we do not have to set up connections to other sources. However, creating an appropriate and large scale vocabulary corpus would be extremely difficult. The latter approach, *direct source-to-target approach*, is to conduct ontology translation directly from a source ontology to a target one, without adopting any kind of interlingua. OntoMorph [223] serves as a typical example of this approach, providing a powerful rule language to represent complex syntactic transformations and a rule interpreter to apply them to arbitrary knowledge representation language expressions.

7.4.2 Overview on Semantic Interoperation Methods

Approaches previously mentioned provide a general approach in creating semantically enriched sources and present general strategies to interoperate with heterogeneous sources. For example, a global schema *merged* from a set of individual sources can provide a reconciled, integrated and virtual view of the underlying sources — actually a (virtual) global schema is set as a *default* assumption in integrating data across federated databases.

In P2P-based digital libraries, however, these strategies may not be simply 'borrowed' because of the special features P2P networks hold, eg. decentralization, dynamism (peer joining and leaving) and autonomy. Hence, to enable access and interoperation between multiple ontologies in semantically sound manners, further investigation is highly required. Herein, a continuum of aforementioned semantic interoperation methods is illustrated in Figure 7.7.

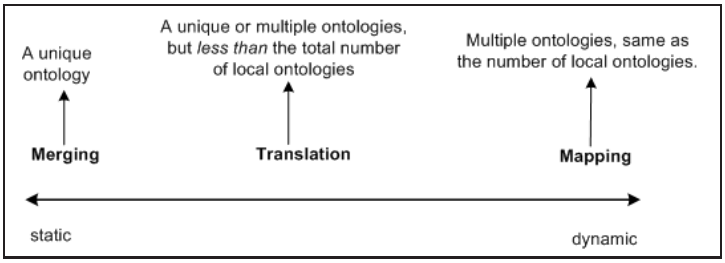

Figure 7.7: The Continuum of Semantic Interoperation Methods

Reasons for making such a continuum are: 1) for better clarifying and locating our strategies in P2P setting; 2) these strategies are not absolutely unrelated (eg. translation based approach can be regarded as *two* separated mapping processes).

On one extreme of the continuum, a single merged ontology acts as a mediator for interpreting specific source ontologies/schemas. As mentioned in Section 5.2.2, approaches concerning global ontology/schema can be rendered in two ways, namely, Global-as-View or Local-as-View approaches. The GaV-based approach, although it is convenient to reformulate queries according to source schemas, is no longer able to support the tasks envisaged by large-scale distributed environment, such as dynamic P2P systems. That is, it is infeasible to generate a *huge* global ontology catering for a large number of semantically heterogeneous metadata standards. In contrast, in LaV-based approach which does not rely

on the global ontology[4], one of the challenging issues in LaV-based approach is to answer queries posed to the global ontology: queries over the global ontology *must* be reformulated in terms of a set of queries over the sources.

On the other extreme of the continuum, the process of ontology mapping does *not* assume any global ontology. Instead, all participating nodes can be loosely coupled and achieve a certain degree of autonomy. Furthermore, any extension to one ontology will not affect the other ontologies which have already been included in the system. This approach seems appropriate for the application scenario of P2P system, however, there is a critical challenging issue that mapping heterogeneous ontologies is often expensive and conducted *offline* by ontology editing tools. It is mandatory to solve this issue before information searching can be conducted in P2P systems.

In the middle of the continuum is the translation-based approach which intends to combine both of the features of ontology merging and mapping. To achieve it, source ontologies on one hand should be supported so as to maintain peer's autonomy; and on the other hand, a set of controlled vocabularies are used to 'speak the same language' across source ontologies. More complicatedly, a virtual ontology or upper level ontology can be further used or extended as a commonly shared conceptual model. Obviously, approaches of this kind can facilitate mappings between source ontologies via shared conceptual model.

To make it brief and clear, we summarize comparison results for these methods in Table 7.1.

From the discussion above, we conclude the adaptability of these three methods in P2P settings as follows:

- Ontology Merging-based approaches hold some merits, such as excellent scalability and low computation cost. However, it is not suitable for applications where dynamicity and flexibility are unavoidable since frequent update of commonly shared ontology/schema will significantly impair low computation cost. In addition, it is also widely agreed that tremendous efforts are required to create such a commonly shared ontology.

- Ontology Translation-based approach looses the requirement on a unique and commonly shared ontology/schema, replacing it with a

[4]However, LaV-based approach does assume a global schema 'integrated' from source schemas, cf. Section 5.2.2.

set of controlled vocabularies with background knowledge. Provided that relations are established between source/target ontology and different parts of the background knowledge respectively, the relation between source and target ontologies can thus be generated. The limitation in this approach is that many practical ontologies are rather semantically 'lightweight' and thus do not carry much logical formalism with them[196].

- Ontology Mapping-based approach fits naturally with the communication profile of P2P-based systems, but it is also expensive to conduct mapping processes, which may put a limit on system *scalability* and would in turn impair the potentials of P2P networks. To alleviate such a problem, efficient mapping mechanisms are needed, especially in aspect of parsing and processing semantic data/information.

7.5 Chapter Achievement

- Semantically enriching metadata is justified as an important phase in realizing semantic searching across heterogeneous sources.

- Process for developing ontological knowledge sources has been presented, namely, Ontology creation/import, enriching metadata with semantics, searching semantic information and using semantic information.

- Conventional approaches for interoperating heterogeneous sources, ie. merging, mapping and translation, have been introduced. These approaches have been further discussed and compared, with the purpose of studying their adaptabilities in P2P settings. The conclusion is that there exists no all-purpose solution while system requirements should be studied before applying specific solutions. For example, if dynamicity and flexibility is highly demanded in P2P communications, the merging based approach should not be adapted.

Table 7.1: Comparison of Semantic Interoperation Approaches in P2P Setting

Feature	**Merging-based App.**	**Mapping-based App.**	**Translation-based App.**
Global Ontology	One	-	One (virtually)
Flexibility	Low (A direct mapping to the global ontology is mandatory; global ontology dependent)	High[a]	Moderate(A virtual global ontology is composed of multiple standardized source ontologies)
Reusability	Low (Focus on specific domains)	Moderate(Ontology mapping algorithm can be reused across the ontologies)	High (Standardized ontologies/metadata schemas are highly resusable)
Dynamicity	Low (Global ontology must be updated when peers leave and join)	High (not affected by peer's leaving and joining)	Moderate (Global ontology and translation rules must be updated when new ontology is added/removed)
Scalability	High (In specific domains where contexts are highly relevant)	Low (when a large number of ontologies are involved, system performance will decrease because of huge cost on mappings)	Moderate
Computation Complexity	Low	High	Moderate
Semantic Completeness	Moderate	High	Moderate

[a]Additional mapping is required when searching sources created heterogeneous schema; no global ontology is involved

Chapter 8

Semantic Relations Elicitation

From the previous chapter, ontology mapping turns out to be appropriate in P2P-based computing environment, along with research challenges in enabling online and efficient mappings. This chapter first proposes a general process for supporting runtime semantic search. Within such a process, we identify that a critical step is to extract relationships between heterogeneous ontologies efficiently and effectively. Typical semantic relations (ie. Semantic Bridges) will be discussed in this chapter and a mechanism of logic-based reasoning will be presented for eliciting semantic relationships. Finally, we test the feasibility of run-time ontology-mapping based semantic search method.

8.1 A Process for Enabling Semantic Search in P2P Network

A process is illustrated in Figure 8.1 for enabling semantic search in P2P network. The purpose of this process is to achieve real-time mapping and query reformulation in practical applications, since one peer can not expect beforehand the metadata schema adopted by incoming queries. Three phases have been designed for the search process: (1)pre-processing; (2) semantic elicitation; (3) caching results.

In the phase of *Pre-processing*, cached mapping results or pre-defined rules are checked whether available. If mapping results have already been recorded, query reformulation can be conducted instantly and the phase

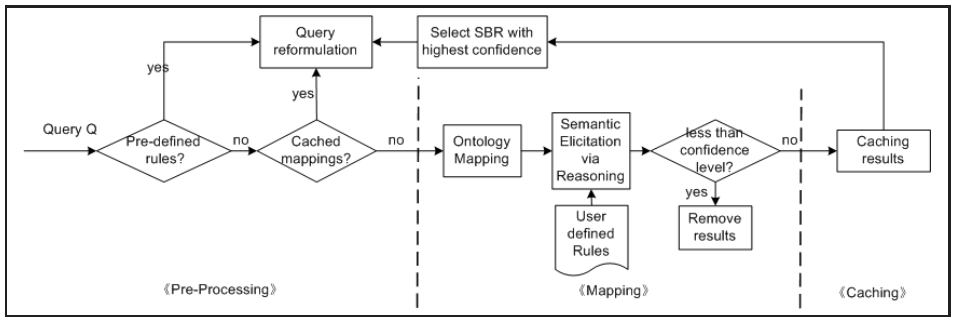

Figure 8.1: The Mapping Process

of mapping will be bypassed.

The *Mapping* process determines the efficiency and effectiveness of query reformulation. After receiving a query q and no cached mappings are found, a new mapping process is triggered for eliciting relations between two ontologies. Additional information reasoning based on pre-defined rules can also be applied during this process, with a goal to achieve better precision (to be discussed in Section 8.5).

In the phase of *Caching*, in order to make mapping more efficient, our searching process caches tuples $< s, t, Sim(s, t) >$ after mapping is completed(via Wordnet) or external rules are inserted (eg. by domain experts, cf. Section 8.5). When a user performs frequent user ontology mappings, his partial user ontologies stored in the cache helps save mapping time. The size of caching the partial ontology mapping results is defined by domain administrators. Different peers may be assigned different amount of caches. For example, super peers could have a large cache while client peers are allocated with a small cache.

Efficient and accurate mappings *from* heterogeneous terms in user queries *to* those in source ontologies act a significant role in achieving *run-time* searching. *Hence our focus is to achieve intelligent and proactive searching services provided that response time is 'endurable' in P2P-based computing environment (eg. less than 5 seconds).*

8.2 General Definitions and Hypotheses

8.2.1 Meanings of 'Run-time'

In our approach the meaning of *run-time* in ontology mapping is two-fold. On one hand, ontology mapping should be conducted online instead of offline since the mapping process is instant and dynamic. On the other hand, *efficiency* should be considered because user queries will be simply abandoned if large latency is occurred during processing them.

Note that the mapping performed in this approach is *partial*, handling only concepts appeared in user queries. Conventional mapping tools usually work on all concepts and in an offline mode. Instead, our mapping mechanism aims to be faster when source ontologies are huge and only a small number of concepts are used in user queries. Actually, from the perspective of online searching, it has been observed that most people use only two word phrases in search engines[1]. It would be a waste of time and resource to map/align *all* concepts and properties in corresponding ontologies.

8.2.2 Peer Communication Model

Each peer in P2P systems can play two complementary roles: information requestor and provider. Figure 8.2 shows a general peer communication model based on such roles.

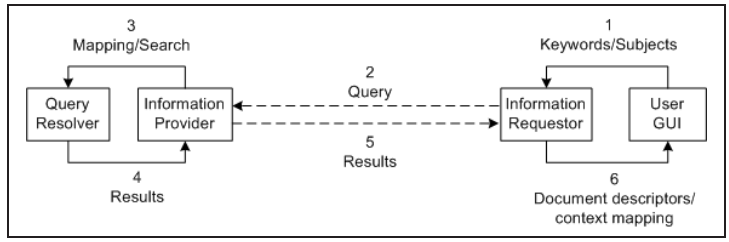

Figure 8.2: The Continuum of Semantic Interoperation Methods

Figure 8.2 illustrates an interaction process between two peers acting as information requestor and provider respectively. The number above the arrows indicates the sequence of interaction. Ontology mapping and

[1]according to OneStat.com: http://www.onestat.com/html/aboutus_pressbox27.html

query reformulation is rendered on phase 3 where the provider will trigger new mapping process if no cached results can be used. Query resolver handles query reformulation and search when mapping process is finished.

8.2.3 Research Hypotheses

Hypotheses for the proposed approach are:

1. Global ontology/schema or predefined controlled vocabularies can not be applied — merging all source ontologies in advance is not feasible because all the possible merging are not foreseeable;

2. Full *autonomy* are expected by all peers in P2P systems. In super-peer systems, only super-peers are regarded to have full independence.

8.3 Semantic Elicitation

The meaning of semantic elicitation is three-fold. Firstly, it minimizes the number of concepts to be mapped between two ontologies such that fewer resources will be wasted on computing irrelevant concepts. It is especially important in situation when large ontologies are involved. Secondly, a set of candidate concepts related to external term is generated. Thirdly, a set of background knowledge about concepts occurring in the two ontologies are applied via logic reasoning.

Many mechanisms are proposed to compare semantic similarity between two strings, for example, longest common substring, longest common subsequence and hamming distance. These mechanisms are based on the *syntactic* meaning (i.e. the spellings) of the two strings instead of *semantic* meanings. Take a simple example: concept author and concept writer are not similar as their morphologies are greatly difference. However, we know that author and concept writer have similar meanings in English. Such a fact reflects that *semantic* meanings of concept names should be considered as well in addition to morphological approaches. To define the semantic meanings of the words, we propose to use the WordNet ontology[197]. WordNet organizes English nouns, verbs, adjectives and adverbs as synonym sets (ie. synset), which are linked by different kinds of relations. Resnik[224] first defined the similarity between two concepts lexicalized in WordNet to be the *information content* of their lowest super-ordinate (lso) (ie. most specific common subsumer). In WordNet,

Semantic Distance $\mathcal{D}$ can be applied to indicate the similarity between two concepts. For example, author and writer are semantically related because they are synonyms in the WordNet ontology and the distance between them $\mathcal{D}$(author, writer) is zero.

To prune irrelevant concepts, we retrieve the semantic distance of their concept names which is smaller than or equal to the semantic distance threshold τ for each pair of concepts between s defined in external ontology O_s and t defined in internal ontology O_t.

$$D(s,t) \leqq \tau$$

which can be further normalized as below:

$$Sim(s,t) = \frac{\tau - D(s,t)}{\tau}$$

The value of similarity $Sim(s,t)$ is ranged from 0 to 1, while a "0" means the compared items are totally different and a "1" means they are identical. The top k pairs of concpets (t,s) with highest similarity degree are selected into a candidate set denoted by Ω. This process is iterated until no terms can be extracted from incoming queries.

In Section 8.4, we discuss possible relations between concepts. Moreover, user-defined rules, ie. background knowledges, can be applied in explicating implicit relations between two relevant ontologies. Further discussion is in Section 8.5.

8.4 Semantic Bridges Between Concepts

Distinct metadata sets can be used for specific domain applications, thus, specifying the relations between relevant or similar terms becomes very important for querying. Inspired by the bridge rule in distributed description logic [225], we study relationships between different conceptual terms and name them as Semantic Bridge Relations (SBR). Based on prior works on associating heterogeneous information resources [197, 226, 135, 227, 225, 228, 229, 152], we summarize SBRs as follows:

- Synonym Bridge: This bridge represents the identical or almost close concepts between different ontologies. For example, the 'PC' and 'Computer' in different ontologies express the same meaning.

- Polysemous Bridge. This bridge represents that the same concept in different ontologies has different meaning. For example, the 'Doctor' in different ontologies may represent a man having PhD degree or a person *doctors* people. Together with synonym bridge, the problem that concept having several meanings and various representations denoting one meaning can be addressed.

- Subsumption Bridge. Processing subsumption is one of Description Logics' major capabilities. This bridge expresses 'broader' or 'narrower' relations between different terms — which are normally named Hypernym Bridge and Hyponym Bridge.

  - Hypernym Bridge. This bridge is represents that one word is more generic or broad than another given word, resulting a possible hierarchy relations of concepts in different metadata sets. For example 'publication' is hypernymic to 'book'.
  - Hyponym Bridge. This bridge is the opposite of the hypernym one and can also be called 'isa' bridge. For example, the 'Student' in an ontology has 'isa' relation with the 'Person' of another ontology.

- Overlapping Bridge. This bridge expresses the terms in different metadata sets are similar but absolutely not identical. One typical example is 'boat' and 'ship' where the difference between them is sometimes blurry. Though these terms do not form an equivalence set, each of them can be precisely defined in specific hierarchy. Yet they are sometimes used loosely and interchangeably in some scenarios by indicating 'related terms'.

- Meronym/Holonym Bridge. This bridge represents part-whole relationships between different ontologies, and is also called 'has a' bridge. One well-known example is the 'Tree' having 'Root', and 'Leaf' of another ontology. Another example in bibliographic world is the 'Thesis' which generally consists of 'PhDThesis' and 'MasterThesis'.

- Opposite Bridge. This bridge expresses that two concepts in different ontologies have opposite meaning, such as 'Man' and 'Woman' in different ontologies. The corresponding constructor of opposite bridge in OWL is 'complementOf'.

- Connect-by Bridge. This bridge represents that concepts in different ontologies can be associated with other terms. For example, 'Professor' of a faculty ontology and 'PhDStudent' of a student ontology can be connected by term 'Supervise'. If the concepts of different ontologies are disjoint on the meaning and can not be connected by some terms, no bridges exist between them.

- Inverse Bridge. The bridge represents that the relations between different ontologies may be inverse. For example, the role of 'teach' in an ontology and the other one 'taughtBy' in another ontology are inverse relations.

In a formal description, a SBR can be written into a tuple:

$$SBR =< b, s, t, c >$$

where b is the hypothesized SBR, ie. semantic relation, between the *external* (ie. incoming) term s in Ontology O_s and *internal* term t in Ontology O_t, while c is the confidence level of trusting the SBR, $c \in [0, 1]$, with 1 indicating full confidence(true) and 0 standing for a false SBR. In this thesis, we consider only synonym ($\equiv$), subsumption bridge ($\sqsubseteq$, $\sqsupseteq$) in order to make our work simple. Support for More complex SBRs, such as *inverse*, *opposite* and *overlapping*, will be considered as future work.

8.5 Semantic Elicitation via Logic Reasoning

8.5.1 Requirements for Describing Complex Relations

Implicitly, ontology-based approaches are based on the assumption that ontology is generally domain-specific. In a domain specific ontology, from a theoretical perspective, only concepts and relationships are described while their relations to external concepts do not need to be presented. Then, in a highly dynamic situation where multiple ontologies instead of a globally shared one are found, the relationships between these ontologies are not defined beforehand. Thus, an extra step, such as ontology mapping mentioned previously, is required to make explicit the dependencies between concepts in different ontologies such that a consensus can be reached between mutually independent digital libraries. In conventional metadata mapping solutions, Description Logics-based languages

(eg. OWL) can use predefined 'constructors' to describe relations between concepts, such as indicating concept A is a 'subClassOf' concept B. However, due to the construction of Description Logics, these constructors are rather limited in expressing relations between properties (i.e. roles), although they are advantageous in describing relations between concepts (e.g., subsumption relation). In addition, *Description Logics are incapable of describing condition based relations.* For example, in some universities, IF a paper is published in the proceedings from a prestigious conference, THEN this paper can be regarded as (ie. 'isa') a journal paper. Roughly, such conditional relations can be regarded as rules that are able to describe complex and implicit relations between ontologies. As noted by Golbreich[230], the advantages of applying rules come from the fact that they can provide various and flexible methods to resolve semantic gaps among ontologies, such as: standard-rules for chaining ontology properties(eg. the transfer of properties from parts to wholes); bridging-rules for reasoning across domains; mapping rules between Web ontologies for data integration; querying-rules for expressing complex queries upon the Web; and meta-rules for facilitating ontology engineering (acquisition, validation, maintenance).

In order to have an explicit approach, we start with some working examples. We use selected portions of two bibliographic ontologies used in two different library collections, *dblp_bib* ontology and *ntnu_bib* ontology, shown in Figure 8.3.

Example A:Both ontologies have a class named *Article*. In the *dblp_bib* ontology, *Article* is a class that is disjoint with other classes such as *Proceedings* and *PhDThesis*. That is, *Article* in the *dblp_bib* ontology includes only articles published in journals. But in the *ntnu_bib* ontology, *Article* consists of all articles released in form of a journal, conference or even thesis. Additionally, the class *Series* does not have correspondent in the *dblp_bib* ontology.

Example A shows that similar terms from two distinct ontologies may have different meanings even if they are obviously derived from the same controlled vocabulary (eg. the Bibtex terminology). Another case for complicated semantic discrepancies is *inheritance*, which allows concepts to be inherited from basic concepts originated in other ontologies.

Example B: Consider two simply ontologies - *dblp_bib* ontology and Dublin Core ontology [64], both of which contain the concept *Publisher*. *Publisher* in *dblp_bib* ontology includes only an organization entity, while in Dublin Core *Publisher* may include a person, an organization, or even

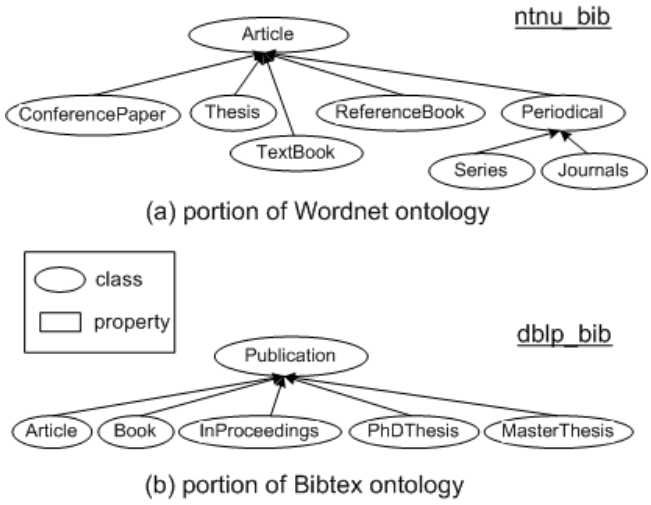

Figure 8.3: Portions of two bibliographic ontologies

a service. Thus, if two other ontologies inherit the concept *Publisher* from the *dblp_bib* ontology and Dublin Core ontology respectively, the semantic discrepancies in *Publisher* between two simply ontologies are thus inherited from the semantic discrepancies between the *dblp_bib* ontology and Dublin Core ontology.

With the goal being to search across heterogeneous digital libraries in a dynamic networking environment, the significance of our work is the approach we use to explicate complex relations between heterogeneous ontologies and provide the least semantic loss. The next section addresses a general process of combining ontologies with rules.

8.5.2 Combining Ontologies with Rules

Keeping in mind that our final goal is to reformulate queries in one ontology to queries in another with least loss of semantics, we come to a process for addressing complex relations between two ontologies. As mentioned in previous sections, relations among ontologies can be composed as a form of declarative rules which can be further handled in inference engines. In our approach, we choose to use the Semantic Web Rule Language (SWRL) [28], which is based on a combination of OWL DL and OWL Lite [154] with the Unary/Binary Datalog RuleML sublanguages[231], to compose declarative rules. Generally, let $\mathcal{S}$ be a SWRL knowledge base, where $\mathcal{L}^c$ is a set

of OWL classes, $\mathcal{L}^r$ a set of relations, and $\mathcal{L}^t$ a set of OWL constants and SWRL variables. A SWRL rule is in the form: $h_1 \wedge ... \wedge h_n \leftarrow b_1 \wedge ... \wedge b_m$, where $h_i, b_j, 1 \leq i \leq n, 1 \leq j \leq m$ are atoms of the form $C(i)$ with $C \in \mathcal{L}^c, i \in \mathcal{L}^t$, or $R(i,j)$ with $R \in \mathcal{L}^r, j \in \mathcal{L}^t$.

The feasibility of using SWRL is two-fold. Firstly, it provides a mechanism for writing formal meaning of ontologies, including rules written in an abstract syntax. Secondly, it allows the adaptation of a standardized expression to explicitly describe relations. However, to the best of our knowledge there is not yet an approach to process multiple ontologies for the purpose of query reformulation. We decided to use the candidate set obtained in mapping process which consists of both source and target ontologies. Actually, it is inspired by application profiles[145] based applications where schemas consist of elements drawn from one or more namespaces, combined by implementers, and optimized for a particular local application. In addition, as an extension to simply syndicating schemas together, domain experts describe relations between similar concepts in form of inferencing rules that can be further processed in inference engines automatically. The entire procedure is illustrated in Figure 8.4.

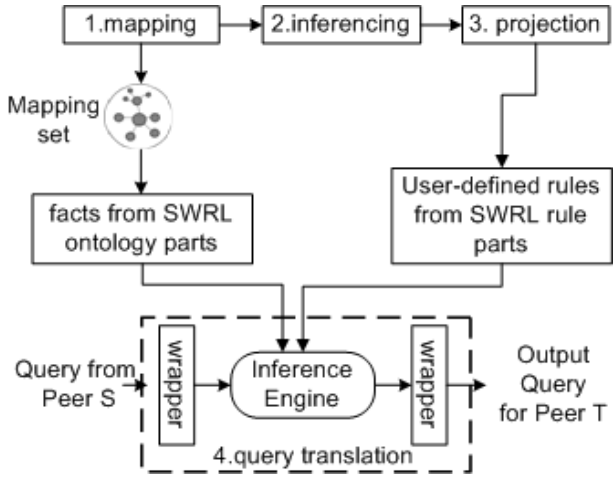

Figure 8.4: Processing Complex Relations

We call the process *logic reasoning based query reformulation*. In this process, we assume there are only *two* different ontologies, namely the source ontology and the target one, since queries are generally submitted from the source peer to the target one. Indeed, any query reformula-

tion involving multiple parties (≥ 3) can be decomposed into multiple pair-wise rewritings, so such an assumption can be neglected in a loose situation. In addition, note that such query reformulation is a process of semantic translation which is more complicated than semantic mapping since indicating corresponding mapping rules require more subtle judgments about the relations between concepts in different ontologies. We break the process into following phases:

1. *Mapping*: combining all available classes, properties and axioms. from both the source and target ontologies with an output of a set of 'mapped' ontologies. The individual namespaces are used as prefixes for duplicated entities respectively.

2. *Inferencing*: it is a manual phase that requires domain experts to inspect relations between related concepts and compose SWRL rules in the form: $h_{target_1} \wedge ... \wedge h_{target_n} \leftarrow b_{source_1} \wedge ... \wedge b_{source_m}$, where inferences rules are drawn from source facts to the target ones.

3. *Projection*: hold SWRL rules which are expressed in the direction from source to target.

4. *Query Translation*: this phase involves transforming facts from the SWRL ontology part and pre-defined rules from the SWRL rule part. Herein, in the input and output interfaces of the inference engine need we a wrapper which implement the *syntactic* translation for the input query and the output one respectively.

One possible problem in reasoning over ontologies via SWRL, as pointed out by Horrocks [232], is that SWRL extends the expressivity of OWL at the expense of the decidability of query answering operations. Fortunately, there are several approaches [233, 234, 230] to cope with this problem. In this thesis, we choose to combine $SHIQ(D)$ [235] and *DL-safe rules* [236] to provide reasoning for ontology translation with respect to query reformulation. When applying *DL-safe rules*, each variables in a rule is required to occur in a non-DL atom in the rule body and thus ensures that each variable is bound to individuals that are explicitly asserted in ABox [236, 237]. We refer readers to [236] to have a full description of the *DL-safe rules* technique. To conduct the reasoning, inference engines supporting $SHIQ(D)$ plus *DL-safe rules* can be applied, eg. KAON2 API[2].

[2]http://kaon2.semanticweb.org/

8.6 Walk-Through Examples

Using **Example A**, suppose the source ontology is *dblp_bib* and the external ontology is *ntnu_bib*. The pre-processing phase will check whether cached mapping information is available for interpreting terms in *ntnu_bib* to corresponding terms in *dblp_bib*. If not, the next phase of semantic elicitation will be applied. Firstly, label-based matchmaking applies algorithms measuring 'distance' between strings (eg. hamming or levenshtein algorithms [238, 239]); Secondly, semantic similarity between concepts and properties are obtained by computing their semantic distance in WordNet. Finally, a candidate mapping set $\mathcal{M}_{s,t}$ is generated by cutting off candidates whose similarity values are less than predefined threshold value (eg. 0.7). After semantic elicitation, mappings $\mathcal{M}_{s,t}$ between these two ontologies could be checked by domain experts manually and particularly, background knowledge — inferencing rules — can be inserted into $\mathcal{M}_{s,t}$. The representation of relations are largely depended on the expert's comprehension and the user's specific requirements, so the final version of $\mathcal{M}_{s,t}$ may not be monotonous. However, the rules and relations defined for concepts must be accepted by participating libraries. In addition to the axioms in two ontologies, all rules have to be expressed in the direction from source to target. A portion of resulting 'knowledge base' for **Example A** is illustrated in Table 2[3].

Query rewriting is conducted in the following steps:

1. Finding Datalog internal rule in the resulting knowledge base and rewriting query;

2. Reasoning based on the merged ontology to infer concepts that are subsumed by the concepts in the query;

3. Rewriting the query in the source ontology to the query in the target one.

Example A.1 (1:1 mapping): Given a query that obtains "all *Journal* papers x published by some one a".

$$Q1(x) \leftarrow Journal(x) \wedge author(x, a)$$

[3]We bind our nomenclatures to that in the First Order Logic where lower case letters from the end of the alphabet stand for variables, lower case letters from the beginning of the alphabet stand for constants and capital letters relations.

Table 8.1: The Resulting Knowledge Base in $< \mathcal{O}_s, \mathcal{O}_t, \mathcal{M}_{s,t} >$

Description Logic Ontologies

$Journal \equiv dblp{:}Article^{+}$
$Journal \sqsubseteq Periodical^{*}$
$Thesis \equiv dblp{:}PhDThesis \sqcup dblp{:}MasterThesis$
$ReferenceBook \equiv dblp{:}Book$
$TextBook \equiv dblp{:}Book$
$Series \not\sqsubseteq dblp.\top$
$author \equiv dblp{:}author$

Datalog rules

$dblp{:}Inproceedings(x, y) \leftarrow$
$ConferencePaper(x), booktitle(x, y)$

Note 1($^{+}$): This mapping is created manually since by label-based mapping, it should be $Article \equiv dblp{:}Article$ which is *not* true after analyzing their implicit meanings in *dblp_bib* and *ntnu_bib* respectively.
Note 2(*): For simplicity, prefixes to concepts and roles in $\mathcal{O}_s$ are not shown.

From Table 8.1, since no rules are related to *Journal*, we thus move to the description logic ontology mapping part. From there we can find a correspondent one-to-one (1:1) mapping: $Journal \equiv dblp{:}Article$, and Q1 can then be reformulated as:

$$Rew\_Q1(x) \leftarrow dblp{:}Article(x) \wedge dblp{:}author(x, a)$$

However, if Q1 is changed into Q1': "all *Periodical* papers (instead of *Journal* papers) x published by some one a" where *Periodical* does not have any *direct* correspondent in *dblp_bib*, the direct mapping strategy will not work. Fortunately, there is a subsumption that indicates *Journal* is subsumed by *Periodical* in *ntnu_bib*, so according to the mapping: $Journal \equiv dblp{:}Article$, a rewritten query exactly as Rew_Q1(x) can be conducted over the *dblp_bib*-created library without affecting the appropriate results that should be returned. That is, the result will not affect *precision* but will affect *recall*. Therefore, in our approach, a warning message will be triggered in this condition.

Example A.2 (1:n mapping): Given a query that obtains "all *Thesis* papers x published by someone a".

$$Q2(x) \leftarrow Thesis(x) \wedge author(x, a)$$

From Table 8.1, we can translate Q2 into Rew_Q2(x) which is basically a *Union* query.

$$Rew\_Q2(x) \leftarrow$$
$$(dblp{:}PhDThesis(x1) \vee dblp{:}MasterThesis(x2)) \wedge dblp{:}author(x, a)$$
$$\Leftrightarrow$$
$$Rew\_Q2(x) \leftarrow (dblp{:}PhDThesis(x1) \wedge dblp{:}author(x1, a)) \vee$$
$$(dblp{:}MasterThesis(x2) \wedge dblp{:}author(x2, a))$$

Example A.3 (m:1 mapping): Given a query that obtains "all *TextBook* or *ReferenceBook* x authored by someone a".

$$Q3(x) \leftarrow$$
$$(TextBook(x) \wedge author(x, a)) \vee (ReferenceBook(x) \wedge author(x, a))$$
$$\Leftrightarrow$$
$$Q3(x) \leftarrow (TextBook(x) \vee ReferenceBook(x)) \wedge author(x, a)$$

From Table 8.1, both of *TextBook* and *ReferenceBook* are subsumed by *dblp:Book*, a new query can be obtained.

$$Rew\_Q3(x) \leftarrow (dblp{:}Book(x) \vee dblp{:}Book(x)) \wedge author(x, a)$$
$$\Leftrightarrow$$
$$Rew\_Q3(x) \leftarrow dblp{:}Book(x) \wedge author(x, a)$$

As different from **Example A.1**, the rewriting procedure of finding a *superclass* in the target ontology will not affect recall (more answers will be returned) but will affect precision. A warning message will be provided in this condition as well if Rew_Q3(x)s is rendered.

Example A.4 (1:0 mapping): Given a query that obtains "all *Series* x authored by someone a".

$$Q4(x) \leftarrow Series(x) \wedge author(x, a))$$

From Table 8.1, no corresponding mappings or subsumption can be found for *Series* to deduce related terms in target ontology *dblp_bib*. It thus results in that the rewriting process can not be finished, and query Q4 has to be discarded and warning message should be generated.

Example A.5 (rule-based reformulation): Given a query that obtains "all conference proceedings' names y where someone a has published papers"

$$Q5(z) \leftarrow author(x, a) \wedge ConferencePaper(x, y) \wedge booktitle(y, z)$$

From Table 8.1, a Datalog intentional rule as follows is matched in part of the query Q5:

$$dblp{:}Inproceedings(x, y) \leftarrow ConferencePaper(x), booktitle(x, y)$$

Therefore, $ConferencePaper(x, y) \wedge booktitle(y, z)$ is substituted by $dblp{:}Inproceedin$ in query Q5; then we can infer $author(x, a)$ into $dblp : author(x, a)$ by reasoning the merged ontology. Finally, query Rew_Q5 is generated as follows:

$$Rew\_Q5(z) \leftarrow dblp{:}author(x, a) \wedge dblp{:}Inproceedings(x, z)$$

8.7 Evaluation

Experiments are conducted to evaluate the effectiveness of run-time ontology mapping. We did not evaluate the performance of semantic elicitation via logic reasoning although reasoning functionality is implemented. Two basic reasons are: firstly, the quality of the rules to be provided by users is unpredictable; secondly, reasoning capabilities of current inference engines (ie. capability for processing SWRL) are limited. Moreover, we are interested in the general computation speed and accuracy of ontology mapping results, so in the evaluation more efforts are put in automatic ontology mapping. Accordingly, a well-known instance-based ontology mapping tool — the GLUE project[240] is employeed for comparison purpose.

The nature of the existing ontology mapping tools is different from our matching mechanism. A fair comparison on the performance is difficult to carry out. Our mapping mechanism performs mappings for concepts appear in the incoming queries while existing ontology mapping tools perform mappings for all concepts and training instances. Our matching mechanism must be faster if the source ontologies are huge and only a small number of concepts are used in the request instance. Moreover, our design caches historical mapping records and partial user ontologies. If a cached record is found during mapping, it even fastens our mapping mechanism. As it is difficult to perform fair comparisons, our experiments should be carefully designed. The factors that we have considered to be evaluated are listed in the following section. Evaluation results are going to be present in the last part of this chapter.

8.7.1 Evaluation Settings

Evaluation Methodology

Our objective is to design an effective ontology mapping mechanism for P2P computing environment. The effectiveness can be measured by *execution time*, *consumed memory* and *accuracy of the mapping results*. The execution time is measured in unit of *second*. Consumed memory means the memory used to store source ontologies, mapping rules, the cached records and the instances (required in GLUE). To have more useful measures that have a fixed range and are easy to compare, we borrow from Information Retrieval [241, 242] the terms of *Recall* and *Precision* which are used to evaluate retrieval performance. Similarly, consider an example mapping request Q of a reference collection (which is all possible mapping pairs found in two ontologies) and its set of R of relevant mapping pairs. Let $|R|$ be the number of mapping pairs in this set. Assume that a given ontology mapping strategy (which is being evaluated) processes the mapping request Q and generates a document answer set A. Let $|A|$ be the number of mapping pairs in this set. Further, let $|R_a|$ be the number of mapping pairs in the intersection of the sets R and A. The *Recall* and *Precision* measures can be defined as follows:

- **Recall** is the fraction of the relevant mapping pairs (ie. the set R) which has been retrieved, ie., $Recall = \frac{|R_a|}{|R|}$
- **Precisoin** is the fraction of the retrieved mapping pairs which is relevant, ie., $Precisoin = \frac{|R_a|}{|A|}$

To have a single measure, the harmonic measure F of *Recall* and *Precision* [2] is applied as follows:

$$F = \frac{2}{\frac{1}{Recall}+\frac{1}{Precision}} = \frac{2 \times Recall \times Precision}{Precision+Recall}$$

Source Ontologies

Three sets of ontologies are used in the evaluation while each set has two ontologies. The first set of ontologies is *small* ontologies containing general terms about 'time'. The second set contains relatively larger ontologies specifying 'bibliographical' knowledge and the third set is within the domain of "tourism". These sets of ontologies are chosen to show that our matching mechanism suits different kinds and scales of ontologies. All

Table 8.2: Two ontologies in 'Time'

	Time.daml	Time-Entry.owl
Concepts	3	16
Properties	4	47
Language	DAML	OWL
URI	http://www.ai.sri.com/daml/ontologies/sri-basic/1-0/Time.daml	http://www.isi.edu/~pan/damltime/e-entry.owl

Table 8.3: Two ontologies in 'Bibliography'

	bibtex.owl	publication.owl
Concepts	15	12
Properties	40	30
Language	DAML	OWL
Namespace	http://visus.mit.edu/bibtex/0.1/bibtex.owl	http://ebiquity.umbc.edu/ontology/publication.owl

ontologies used for evaluation are found in the Web, while some of them are composed in different languages, such as DAML. We adopt OWL as a standard ontology language and convert those not written in heterogeneous language into OWL. Table 8.2, 8.3 and 8.4 show corresponding information about the ontologies used.

Implementation Details

The matching mechanism has been implemented using Java (Java SDK v5.0) language and OWLAPI[243] and KAON2[244]. The API of KAON2 is capable of manipulating OWL files and providing novel algorithm (ie. reducing a SHIQ(D) knowledge base to a disjunctive datalog program) for reasoning.

A normal desktop computer is used as the simulator of query reformulation platform. It executes the mapping mechanism, reasoning, storing domain ontology, caching mapping results and partial ontologies. The mapping mechanism takes one source ontology and incoming query as inputs. In practical experiment, we map source ontology and external on-

Table 8.4: Two ontologies in 'Tourism'

	travel1.owl	travel2.owl
Concepts	34	52
Properties	10	107
Language	OWL	OWL
Namespace	http://protege.stanford.edu/plugins/owl/owl-library/travel.owl	http://protege.stanford.edu/plugins/owl/protege (local)

```
ontology1 = D:/Project/mapping/ontologies/bibtex1.owl;
ontology2 = D:/Project/mapping/ontologies/bibtex2.owl;
extract_threshold = 0.7
labelonly_weight = 0.3
distance_weight = 0.5
datatype_weight = 0.1
cardinality_weight = 0.1
mappingFile = D:/Project/mapping/ontologies/result.txt;
reasoning = NO;
```

Figure 8.5: Configuration File

tology, instead of designing specific queries and parsing queries. Herein, we anticipate that the processing time on ontologies themselves is acceptable. To evaluate the effects brought by reasoning, we focus on the 'execution time' instead of 'accuracy' because inferencing rules are created by domain experts and are assumed to be more precise than automatic mapping functions. Note that the time spent on investigating and setting up rules are unpredictable and not counted in the total execution time.

A configuration file is used to define initialization parameters. All parameters are provided by users and tunable. Herein, we put a threshold value as 0.7 to filter out 'unrelated' mapping pairs. To compare the semantic similarity, we assign a weight of 0.5 to the word distance value. Additionally, we consider morphological information, such as 'label', 'datatype' and 'cardinality', and in the sample configuration file we have assigned different weights (ie. 0.3, 0.1 and 0.1 respectively) to them (cf. Figure 8.5). The concepts presented in the queries are matched with the source ontology. A matching table with the concepts in the incoming queries and the concepts in the second ontology are outputted to predefined file (ie. "mappingFile").

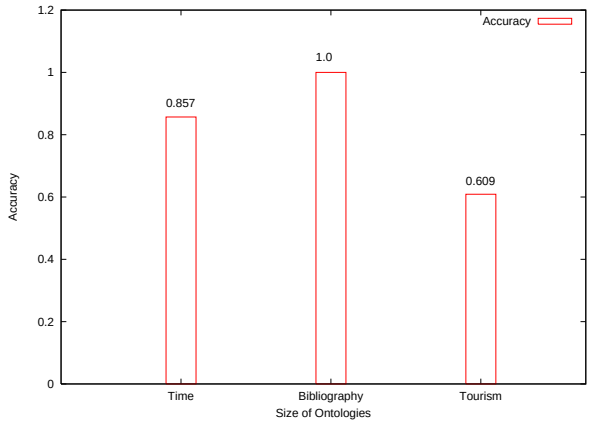

Figure 8.6: Accuracy

8.7.2 Evaluation Results

Three experiments are conducted by using three sets of ontologies. The sizes of ontologies are from small (eg. Time.owl) to moderate (eg. travel2.owl) and thus full mappings between ontologies are conducted. As we have anticipated, the experiment results are promising, achieving acceptable accuracy and keeping execution time within sub-second level. The accuracy is averagely more than 80% (cf. Figure 8.6). Specifically, two ontologies in domain of 'Bibliography' achieve best results, finding all relations between concepts and properties. The reason is that these two ontologies have a great similarity degree, especially from the name of labels for respective ontologies. In contrast, when the size of ontologies increase, some irrelevant concepts/properties are found related. Take the 'Tourism' ontologies for example, 'coach' is wrongly found related to 'beach', in addition to the correct relationships found between 'coach' and 'car'.

One of the comparable approach is the GLUE project led by Doan[240]. GLUE achieved accuracy ranging from 66 to 97%. However, when processing ontologies of similar sizes, GLUE took 10 to 183 seconds to accomplish a mapping process, as derived from the evaluation results in [245]. In contrast, our design needs roughly 10 times less than in GLUE, ranging from 1 second to 5 seconds (cf. Figure 8.7). One of the major reasons is that GLUE applies machine learning techniques which require extra phase in computing instances to train its learner besides comparing the taxonomic structure of the ontologies.

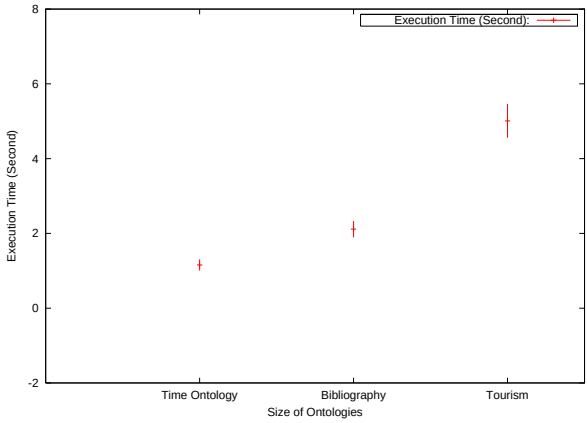

Figure 8.7: Execution Time

Experiments have also been conducted over rule-based reasoning, basing on KAON2 inference engines. If taking into consideration only the time in processing rules, we can have promising response time by running them in KAON2.

Our mapping mechanism has achieved averagely more than 80% accuracy within acceptable execution time. That is, it is feasible to apply it in search in highly distributed environment. Although the GLUE system is slightly more accurate than our system by comparing instances, it is more than 100% slower. Setting up rules in our design requires human intervention in order to realize better mapping results. It is expensive to conduct this phase though. So in situations where execution time is highly concerned, this phase can be dropped.

The bottleneck for our mechanism is still at parsing ontologies. In the above experiments, the larger the source ontologies, the slower are the parsing. We have attempted to find a better parser. However, only several parsers, such as Jena[246] and OWLAPI[243] can support most of the features proposed by OWL. This is also a major reason we did not consider instance-based ontology mapping as is reported to have higher precision[240], because there will be a large overhead in parsing instances, especially when extracting information from large ontologies. Even if instances need to be parsed, such as machine learning-based approach, we recommend that the number of instances to be processed be minimized and concepts having relationships and properties be avoided. The most

effective solution is that a more efficient ontology parser should be designed.

Other optimized approaches, such as caching, can be employed to speed up the process of obtaining mapping results. One example is distributed Java Caching System (JCS), which provides a means to manage cached data of various dynamic natures, supporting high read, low put applications.

8.8 Chapter Achievement

- Presented a general semantic search process in P2P settings, basing on ontology mapping approach.
- Presented requirements for describing complex relations in order to deduce implicit semantic information which can not be handled by ordinary mapping mechanism.
- Evaluation has been conducted on testing our design and mechanism, justifying the feasbility of ontology mapping-based approach.

Chapter 9

Prototype Implementation

This chapter introduces the prototype system we have implemented so far. Our *purpose* is to justify the feasibility of the proposal, such as real-time communication between peers and query response time in JXTA-based digital library systems. We thus focus mainly on justifying functionalities rather than technical implementation details. The part concerning run-time query reformulation according to instantly generated mapping results is left unimplemented in this prototype because of programming workload and lack of annotated records. However, these functionalities have been considered when designing the prototype system, and interfaces have been provided for further extension.

9.1 Prototype Architecture

The prototype architecture is composed of five major parts and Figure 9.1 illustrates the architecture of a generic peer. Note that these parts are presented from a conceptual model perspective, rather than practical constituents (eg. objects) in the prototype system.

- **Local Original Sources (LOS)**: Private person may have varied local repositories to store personal information. These information could be harvested from different data providers and be in different formats, such as structured database, unstructured textual file or semi-structured XML files. The component LOS is responsible for storing the information and more importantly, converting them into XML formatted files. That is, LOS provides an interface to have standardized semi-structured XML files for further processing.

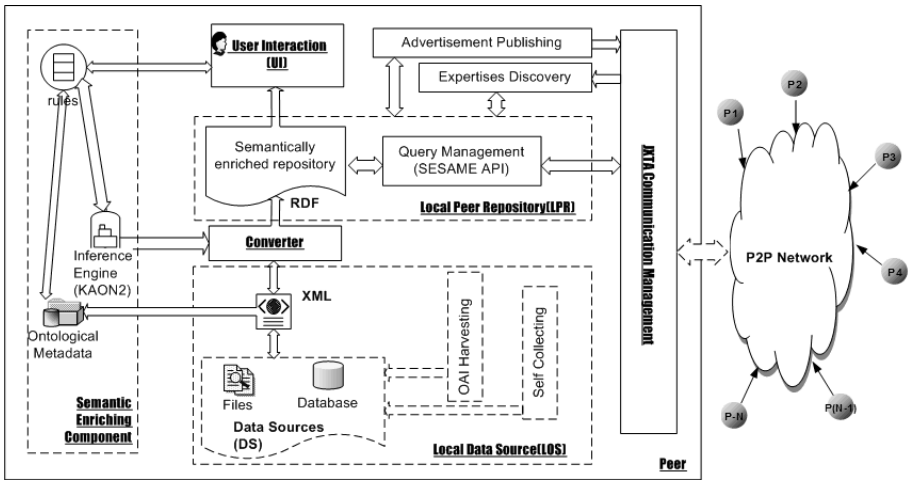

Figure 9.1: General Peer Architecture

- **Converter**: The component Converter is responsible for extracting and re-organizing information from LOS and transforming results to LPR. Along with such process, converted records are formatted in RDF/OWL with semantically enriched information in line with available ontologies. The LPR is expected to have semantics-enriched information to be shared with other peers.

- **Local Peer Repository (LPR)**: LPR is the real resource for the external information searching and internal user browsing and navigating. Nevertheless, strictly speaking, this component is not only an ontology-based resource but a hierarchical- and knowledge-based repository. Additionally, SESAME[8], a set of API for managing RDF files, is used to store and query over local repository, providing semantic search functionality for the system.

- **User Interaction (UI)**: In this component, users can import records in RDF into local repository and export returned results to local system. In UI interface, user can browse and edit mapping results, and even insert inferencing rules. UI also provides functionalities that allow for publication of special services, but also the sending of requests to a crawler for external information collection.

- **JXTA Communication Interface**: We use in this prototype the

JXTA framework for constructing a super-peer network and enabling communication between peers.

- **Semantic Enriching Component (SEC)**: This component is responsible for not only semantically enriching metadata records in the process of converting, but also providing supports in creating, storing and reasoning user-defined inferencing rules.

In a more direct and concise manner, we divide the interactions in the prototype into four layers(cf. Figure 9.2), ranging from lower physical communication to upper 'semantic' operations.

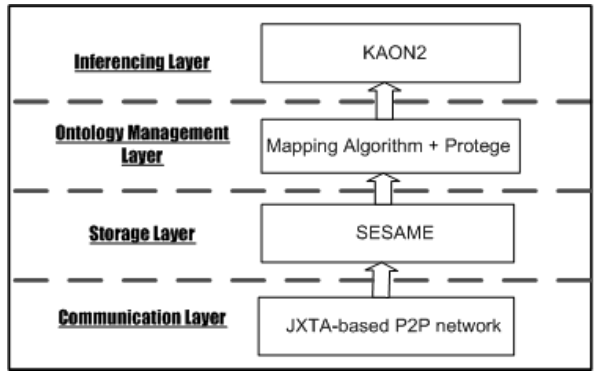

Figure 9.2: System Communication Layer

At the bottom of Figure 9.2 is the physical communication layer which is based on former JXTA platform we have developed [247]. The second layer - storage, uses SESAME API [8] to store metadata records and process queries in form of RDF repository. This layer includes both of LOS and LPR parts shown in Figure 9.1. In the layer of Ontology Management, specific ontology mapping algorithm can be applied for interpreting incoming queries into the one formatted by local format. At the top layer, rules can be created manually or by using ontology editing toolkit, such as Protege OWL Plugin[1]. The functionalities of SEC are realized at this layer. Finally, inference engine, such as KAON2, can be used to handle the reasoning task.

[1]http://protege.stanford.edu/plugins/owl/swrl/index.html

9.2 UML Diagram of Prototype

The upper level UML Class diagram is illustrated in Figure 9.3.

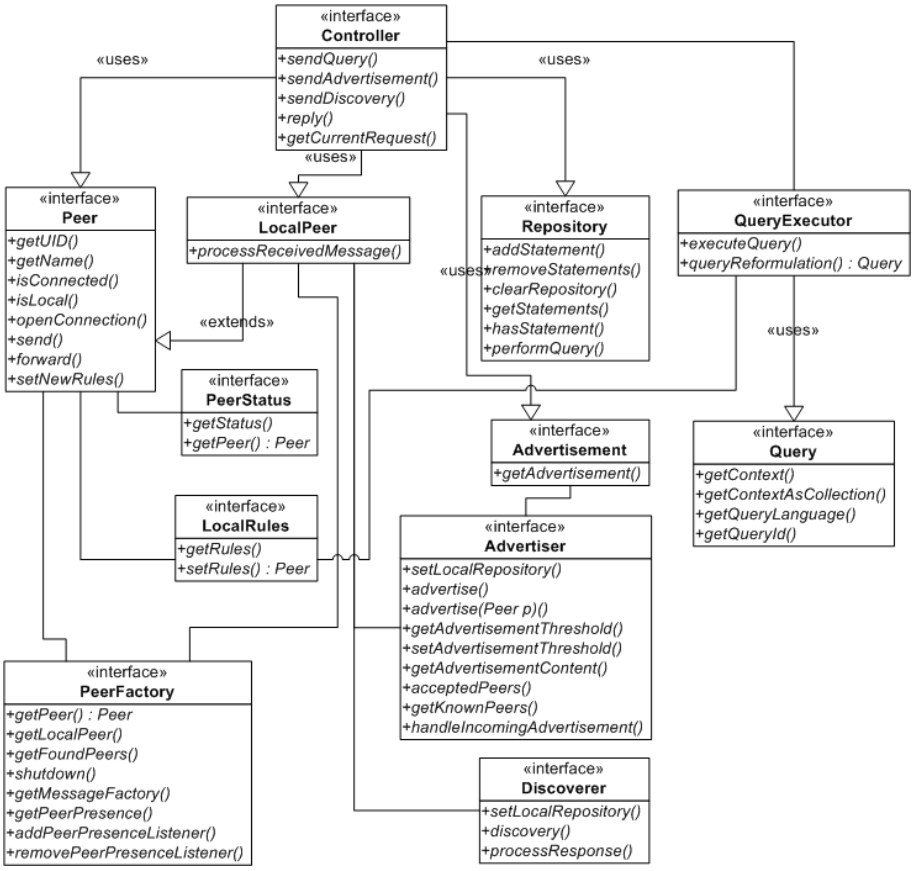

Figure 9.3: Upper Leverl Class Diagram (Interfaces)

Figure 9.3 presents the upper level relationships among different classes. For brevity, this diagram omits most 'less important' and utility classes. The major objects are *Peer Repository* and *Query*. All these objects are directly or indirectly accessible by object *Controller*. Instance *Peer* is 'generated' from *PeerFactory* and may have *status*, namely 'connected', 'not connected', 'local', 'dead' and 'unknown'. Real-time mapping, and rule inferencing are left unimplemented, but corresponding interfaces have been created for further development. For examples, class *Rules* and class

QueryExecutor.

9.3 Adopted Technologies

9.3.1 JXTA Framework

JXTA [118] is a P2P interoperability framework created by Sun Microsystems. It provides the minimal requirements for a generic P2P network, stripping it of all the policy-specific logic and components. This leaves only the building-block constituents that almost all P2P applications can use, regardless of their intended users and specific implementation. In other words, the JXTA components enable and facilitate the simple fabrication of P2P applications without imposing unnecessary policies or enforcing specific application operational models.

The core building blocks for JXTA framework are:

- *Peers and peer groups*: A peer group is a collection of peers that share resources and services.
- *Services*: JXTA services are available for shared use by peers within a peer group. In fact, a peer may join a group primarily to use the services available within that group.
- *Pipes*: A pipe instance is, logically speaking, a resource within a peer group. It forms one way to transfer data, files, information, code, or multimedia content between peers. JXTA pipes are used to send messages (with arbitrary content) between peers.
- *Messages*: JXTA messages are data bundles that are passed from one peer to another through pipes, including segments of header, source and target endpoint information, and message digest.
- *Advertisements*: The content of an advertisement describes the properties of a JXTA component instance, such as a peer, a peer group, a pipe, or a service. For example, a peer having access to an advertisement of another peer can try to connect directly to that other peer. A peer having access to an advertisement of a peer group can use the advertisement to join that group. The current Internet analogue to an advertisement is the domain name and DNS record of a Web site.

An illustration of how peer discovers and joins a 'peergroup' is shown in Figure 9.4. All peers can publish their profiles (i.e., content summary) in way of 'advertising'. One peer in JXTA can thus discover other peers by discovering posted 'advertisements' and then join favorite peer groups. Communications between peers are conducted by 'pipes' specifically generated by them.

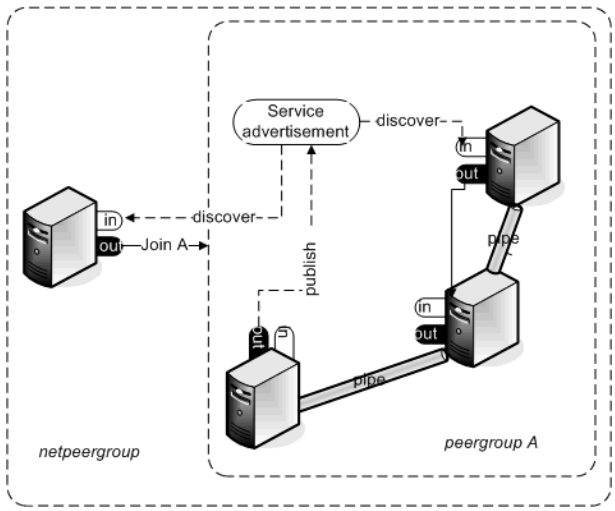

Figure 9.4: Peer Discovering and Joining

9.3.2 SESAME

Sesame[8] is an open source Java framework for storing, querying and reasoning with RDF and RDF Schema. It can be used as a database for RDF and RDF Schema, or as a Java library for applications that need to work with RDF internally. For example, suppose one need to read a big RDF file, find the relevant information for his application, and use that information. Sesame provides necessary tools to parse, interpret, query and store all this information.

In Figure 9.5, an overview of Sesame's overall architecture is given.

At the bottom of Figure 9.5 is the Storage And Inference Layer(SAIL) API, acting as an internal Sesame API that abstracts from the storage format used (i.e. whether the data is stored in an RDBMS, in memory, or in files, for example), and providing also reasoning support. SAIL implementations can also be stacked on top of each other, to provide functionality

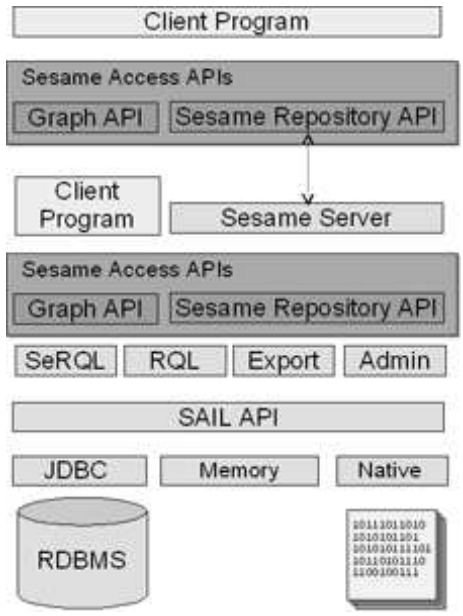

Figure 9.5: The SESAME architecture, from [8]

such as caching or concurrent access handling. Each Sesame repository has its own SAIL object to represent it. On top of the SAIL are Sesame's functional modules, such as the SeRQL, RQL and RDQL query engines, the admin module, and RDF export. Access to these functional modules is available through Sesame's Access APIs, consisting of two separate parts: the Repository API and the Graph API. The Repository API provides high-level access to Sesame repositories, such as querying, storing of rdf files, extracting RDF, etc. The Graph API provides more fine-grained support for RDF manipulation, such as adding and removing individual statements, and creation of small RDF models directly from code. The two APIs complement each other in functionality, and are in practice often used together. The Access APIs provide direct access to Sesame's functional modules — a client program (for example, a desktop application that uses Sesame as a library), or the next component of Sesame's architecture, the Sesame server. This is a component that provides HTTP-based access to Sesame's APIs. Then, on the remote HTTP client side, we again find the access APIs, which can again be used for communicating with Sesame, this time not as a library, but as a server running on a

remote location.

9.3.3 KAON2

KAON2[244] is an industry strength reasoner for OWL ontologies. The following functionalities are provided by KAON2:

- KAON2 provides an integrated API for reading, writing, and management of OWL DL ontologies extended with SWRL rules. Currently, OWL RDF and OWL XML file formats are supported.
- KAON2 provides a built-in reasoner for OWL DL (except nominals and datatypes), extended with DL-safe subset of SWRL. (I.e. KAON2 fully supports SHIQ extended with DL-safe rules.)
- Reasoning is based on novel algorithms, which reduce an OWL ontology to a (disjunctive) datalog program. These algorithms allow KAON2 to handle relatively large ontologies with high efficiency. Its performance compares favorably with other state-of-the-art OWL DL reasoners.
- KAON2 supports the answering of conjunctive queries expressed in SPARQL[248].
- KAON2 supports the DIG interface, and can therefore be used with ontology editors such as Protégé.
- KAON2 can access information stored in relational databases based on mappings between ontology entities and database tables.

9.4 GUI

Figure 9.6 illustrates the main user interface of our prototype system. In Figure 9.6 with a manual peer selection interface. On the left panel, user can limit the search scope, such as local peer, automatic search and selected peers. In this figure, user selects the third option and is able to select specific peers connected. As to the other items on the left, user can conduct simple search, such as that in search engine, and he can also submit advanced search by indicating corresponding values. The returned results are shown on the right panel.

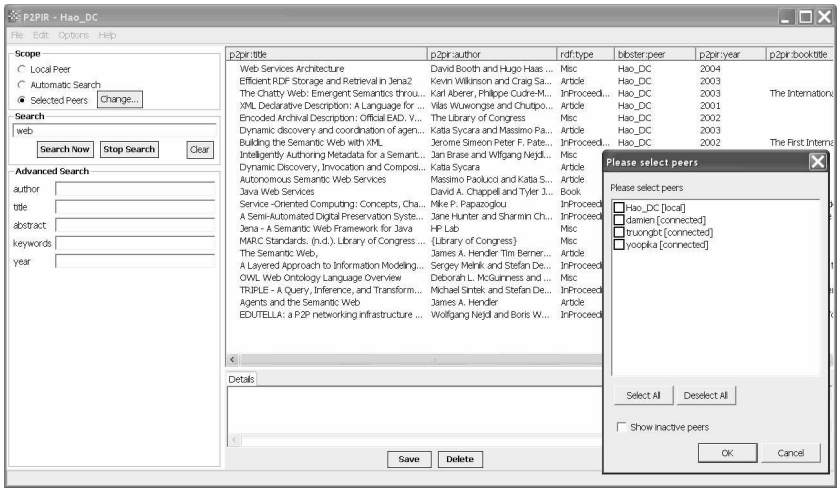

Figure 9.6: The Prototype GUI

Figure 9.7 shows the manual mapping interface we have implemented for rule composing purpose. In this component, user can manually specify the relationships (eg. isA, compose and hasA) between two ontologies (mainly concepts) with different levels of confidences, namely, high, medium and low.

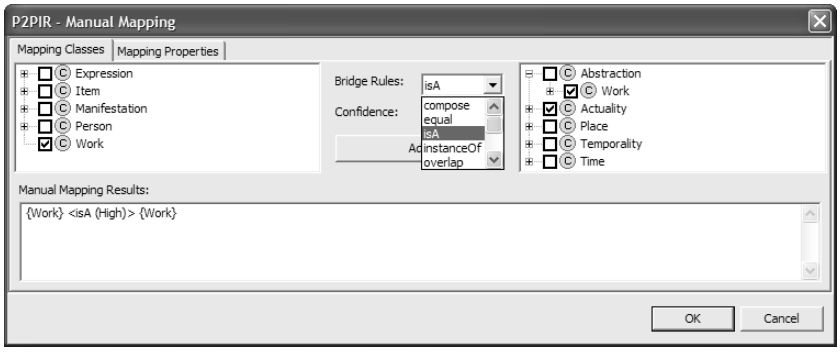

Figure 9.7: The Manual Mapping GUI

9.5 Chapter Achievement

- Presented prototype architecture and abstract model from a general and extensible point of view. The architecture and abstract model can also be reused in constructing P2P-based digital library systems.
- Illustrated upper level class diagram of the prototype system. The class diagram can be widely adopted in applications concerning information search over P2P-based networks.
- Introduced technologies adopted in the prototype.
- Described GUI interfaces which justify the feasibility of conducting real-time searching over P2P-based digital library system.

Chapter 10

Conclusion

This chapter concludes the thesis by summarizing the answers to the research questions. Contributions and future work are to be presented as well.

10.1 Answers to the Research Questions

The main research question, which is presented in Chapter 1, is:

How and to what extent a P2P architecture extended with semantic technologies can enable search of the same quality as if the system was one centralized library?

The main research question has been answered, in general, by the evaluations on the enhanced super-peer model, the ontology mapping mechanism and as well as the design and implementation of the semantic search prototype for P2P-based digital libraries. First, we justified that a super-peer based hybrid infrastructure is appropriate for sharing distributed information among digital libraries. Second, different ontology interoperation methods have been investigated for P2P-based applications while the ontology mapping approach is more suitable for highly dynamic and flexibility-required situations. Finally, a JXTA-based framework has also been applied to implement a super-peer based prototype system to enable semantic search services.

Our answers to the more specified research questions are as follows:

Q1: *How suitable are various P2P infrastructures in decentralized digital library solutions?* Not all P2P infrastructures are suitable for digital

library applications. Different application scenarios, eg. file sharing or global schema-based federated searching, may have different requirements on system architecture. In this thesis, we have proposed a benchmark for guiding users in choosing appropriate P2P infrastructure under specific 'application scenarios'.

Q2: *What kind of metadata interoperation method should be adopted in P2P-based digital library systems?*
As mentioned previously, even in some specific domain such as digital library, different application scenarios may exist. Therefore, we conclude that the selection of metadata interoperation method is also application specific. For example, ontology merging-based approach is suitable for stable environment where high scalability and low computation cost are emphasized. In our work where a dynamic P2P network is concerned, we have chosen to apply the ontology mapping-based approach which provides maximum flexibility in interoperation among peers. In addition, we have designed a semantic search process in P2P setting. Evaluation has shown that this approach is feasible.

Q3:*How suitable semantic technologies (ie. ontologies and inference mechanisms) are in eliciting implicit semantic relations between schemas and supporting search?*
We have also presented the requirements for describing complex relations and candidate inferencing engine, such as KAON2, for reasoning over user-defined rules (eg. in format of SWRL). In addition, we have presented walk-through examples (cf. Section 8.5) to illustrate a process for enabling semantic search via setting up complex relations between heterogeneous ontologies.

10.2 Contributions

The contributions of this work are summarized as follows:

1. An elaborate investigation has been conducted in identifying the strengths and weaknesses of both peer-to-peer and Semantic Web technology. Based on the investigation, we concluded that these two fields are complementary, rather than mutually exclusive. There are great advantages to be gained by combining them in conducting semantic searches in a large-scale distributed environment.

2. A tentative benchmark has been proposed for selecting appropriate peer-to-peer networks for specific digital library construction. In particular, our project has extended classic super-peer-based networks with load-balancing and self-organizing functionalities, thereby catering for dynamic feature assumed in this work. Evaluation results have shown such an extended model is able to cope with continuous departures of peers, overload caused by the joining of peers, or even a system catastrophe.

3. This work has justified the demand for explicit semantics in enhancing both searching precision and recall. A semantic search process has been proposed for facilitating interoperation between heterogeneous ontologies in P2P networks, involving runtime ontology mapping and logic based reasoning.

4. This work has evaluated run-time ontology mapping mechanism for semantic search in peer-to-peer network. Evaluation results showed that run-time ontology-mapping can be achieved in an acceptable time, such that run-time query reformulation can be realized. To our knowledge, little work has been conducted in investigating such issues.

10.3 Limitations and Future Work

This thesis studied two most important technologies nowadays which can bring great effects to the construction of the future digital library systems. However, due to the complexity and breadth of P2P network and Semantic Web technologies, it is almost impossible to cover every aspect which may turn out to be critical. For example, we do not have a running example to demonstrate the search process we have proposed.

Unfortunately large scale experiment has not been conducted in practical applications.

In the current design of our work, we have dropped instance-based mapping in the process of ontology mapping in order to achieve reasonable response time. However, the "cost and benefit" issue has not been researched.

In future work, approaches are still required on lightweight ontology mapping tools and parsers, aiming to make it flexible to suit in different devices.

Another alternative approach for peer-to-peer ontology mappings is to distribute similarity calculations to different resource-rich peers (eg. super peers) in the connected P2P networks. Resource-consuming computation, such as ontology mappings can be distributed to 'idle' peers such that machines are better utilized. Distribution of computation tasks can also help to improve the efficiency of mapping mechanism. If source ontologies are partitioned and distributed to different peers to perform mapping, it can effectively reduce the time used for ontology parsing.

It is also important to note that implementation of RDF, OWL, and the Semantic Web as a whole will be a gradual process. Therefore, the Semantic Web may initially be restricted to intranet and extranet applications until questions about *information security* can be sufficiently addressed.

The digital library of the 21st century will radically transform how we interact with information and knowledge. Traditionally, digitized online information has been dominated by data centers with large collections indexed and stored by trained professionals. The inception of the World Wide Web and the network infrastructures for distributed computing have rapidly developed the technologies of collections for independent communities. In the foreseeable future, online information will be dominated by *small* collections maintained and indexed by individual communities themselves. Under the compelling vision of Semantic Web, future digital libraries will rely on *scalable semantics*, on automatically indexing the community collections so that users can effectively search within billions of repositories. The most important feature of the infrastructure shall therefore be able to support semantic correlation across distributed and heterogeneous collections.

User authentication in peer-to-peer systems, and efficient intelligent mapping in peer-to-peer ontology mappings are the suggested future works. It is believed that there exist many other possible enhancements for applications and systems in such distributed computing environment. With the great efforts from the researchers, distributed computing environment is going to be reality in the very near future.

Appendix A

List of Publications

This appendix lists some of the papers published in conference, workshops, and as well as journals. Some of the research results have already been presented in the thesis. To make it concise and referential, we list them as follows:

1. Hao Ding, Yun Lin, Bin Liu: Towards a Terabyte Digital Library System. IDEAL 2003: 1042-1046.
   ***Abstract**: To access these data quickly and accurately, we are developing a distributed terabyte text retrieval system. To solve the interoperability and extensibility among different information resources, we introduced our solutions of three kinds of metadata schemes. Furthermore, because of the complexity in Chinese language, we made an approach in word segment methods to increase the efficiency and response time of the digital library system. In the testbed, we put an extra layer in the cache server and designed a new algorithm based on VSM. With the query cache, system can search less data while maintaining acceptable retrieval accuracy.*

2. Hao Ding, Ingeborg Sølvberg, Yun Lin: A Vision on Semantic Retrieval in P2P Network, in the IEEE 18th International Conference on Advanced Information Networking and Applications (AINA) 2004: 177-182.
   ***Abstract**: P2P systems are a revival paradigm for information sharing among distributed nodes in the network. Currently, many research projects or practical applications have emerged from the early ICQ, Napster, Gnutella to most recently CAN, Gnougat, etc., but*

few of them support semantic retrieval. The advent of Semantic Web is a highly innovative manner to enhance both the precision and recall simultaneously. This paper investigates a searching problem as encountered in a tourism scenario. Based on the scenario, we introduce several main requirements for constructing semantic retrieval in P2P network. Bared an ambitious goal, we describe a preliminary architecture of average peer. Finally, we offer an approach for a critical part of the architecture — the wrapper, which aims to alleviate the mismatches caused by the content representations among various peers.

3. Hao Ding, Ingeborg Sølvberg: Towards the Schema Heterogeneity in Distributed Digital Libraries, in the Proc. of the 6th International Conference on Enterprise Information System(ICEIS) 2004: 307-312.
   ***Abstract**: In this paper, we discussed the problems brought by the schema heterogeneity in general digital library applications, especially those found in the application of the OAI-PMH protocol. This paper studies the problem from two perspectives, i.e. the schema and the architecture respectively. A preliminary architecture is provided that integrates the ontology, agent, P2P together to support the schema mapping. A semantic negotiation strategy between the heterogeneous agents has also been described.*

4. Hao Ding, Ingeborg Sølvberg: Exploiting Extended Service-Oriented Architecture for Federated Digital Libraries. ICADL 2004: 184-194.
   ***Abstract**: In order to support various requirements from the user's perspective, digital library (DL) systems may need to apply a large variety of services, such as query services for a specific DL, mapping services for mapping and integrating heterogeneous metadata records, or query modification and expansion services for retrieving additional relevant documents. This paper focuses on exploiting an extended Service-Oriented Architecture - Peer-based SOA (PSOA) for DL development with the goal of alleviating the weaknesses in the basic SOA infrastructure, especially in the aspects of scalability and interoperability. We also present our work in how to combine the Semantic Web and Web Services together to support interoperability over heterogeneous library services. A query service example is also presented.*

5. Hao Ding, Ingeborg Sølvberg: Choosing Appropriate Peer-to-Peer Infrastructure for Your Digital Libraries. ICADL 2005: 457-462.
***Abstract**: Peer-to-Peer (P2P) overlay network aims to be a feasible platform for building federated but autonomous digital libraries. However, due to a plethora number of P2P infrastructures and corresponding functionalities, it is often not easy to choose appropriate candidates for specific applications. This paper is devoted for this issue by comparing some typical P2P systems widely used in digital library or database communities and extending an open discussion on how to determine proper infrastructures according to specific system requirements.*

6. Hao Ding, Ingeborg Sølvberg: Semantic Data Integration Framework in Peer-to-Peer based Digital Libraries. Journal of Digital Information Management 2005, Volume 3(2).
***Abstract**: This paper presents our approaches in integrating heterogeneous metadata records in Peer-to-Peer (P2P) based digital libraries (DL). In this paper, the advantages of adapting P2P network over other approaches are to be presented in searching information among moderate-sized digital libraries. Before we present the semantic integration solution, we describe the P2P architecture built in JXTA protocol. By adopting JXTA protocol, peers can automatically discover the other candidates which can provide most appropriate answers. Such feature is realized by the advertising functionality which is introduced in the query process in the paper. As to the metadata integration, since resources may adopt distinct metadata, standardized or non-standardized, we employ the most widely adopted Dublin Core [17] as the globally shared metadata to sponsor the interoperation. This paper also describes the mechanism of applying inference rules to convert heterogeneous metadata to local repository.*

7. Hao Ding, Ingeborg Sølvberg: Rule-based Metadata Interoperation in Heterogeneous Digital Libraries, in the Electronic Library Journal, 2006 (To Appear).
***Abstract**: This paper describes a system to support querying across distributed digital libraries created in heterogeneous metadata schemas, without requiring the availability of a global schema. We investigated the advantages and weaknesses of ontology based applications and have justified the utility of inferential rules in expressing complex relations between metadata terms in different metadata schemas. A*

process is designed for combining ontologies and rules for specifying complex relations between metadata schemas. We collapsed the process into a set of working phases and provide examples to illustrate how to inter-relate two similar bibliographic ontology fragments for further query reformulation. A new approach is proposed for facilitating heterogeneous metadata interoperation in digital library systems as a way of empowering ontologies with rich reasoning capabilities.

8. Hao Ding, Ingeborg Sølvberg: An Enhanced Super-Peer Model for Digital Library Construction, in the Proc. of the 7th International Conference on Web Information Systems Engineering (WISE), LNCS 4255.

   ***Abstract**: Peer-to-Peer (P2P) overlay network has emerged as a major infrastructure for constructing future digital libraries. Among various P2P infrastructures, super-peer based P2P network receives extensive attention because the super-peer paradigm allows a node to act as not just a client, but also serve for a set of clients. As different from conventional file-sharing paradigm, digital library applications have more advanced requirements on system independence/autonomy, robustness and flexible communication. This paper is devoted for constructing digital library systems built upon such super-peer based network, i.e. JXTA framework. Evaluation results are to be presented concerning network initialization, loading balancing and self-organizing.*

Bibliography

[1] "The BIBSYS digital library system." http://www.bibsys.no, 2006.

[2] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. USA: Addison-Wesley, 1999.

[3] W. H. Mischo, "Digital libraries: Challenges and influential work," *D-Lib Magazine*, vol. 11, no. 7/8, 2005. http://www.dlib.org/dlib/july05/mischo/07mischo.html.

[4] M. T. Schlosser, M. Sintek, S. Decker, and W. Nejdl, "Hypercup - hypercubes, ontologies, and efficient search on peer-to-peer networks.," in *AP2PC* (G. Moro and M. Koubarakis, eds.), vol. 2530 of *Lecture Notes in Computer Science*, pp. 112–124, Springer, 2002.

[5] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, eds., *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge University Press, 2003.

[6] "Semantic Web presentations: An overview of the semantic web." http://www.w3.org/Consortium/Offices/Presentations/SemanticWeb/34.html, 2005.

[7] E. Mena, V. Kashyap, A. P. Sheth, and A. Illarramendi, "Observer: An approach for query processing in global information systems based on interoperation across pre-existing ontologies.," in *CoopIS*, pp. 14–25, 1996.

[8] "Sesame: Rdf schema querying and storage." http://www.openrdf.org, 2005.

[9] "Dublin core medata initiative.." http://www.dublincore.org, 2003.

[10] I. H. Witten and D. Bainbridge, *How to Build a Digital Libary.* Morgan Kaufmann, 2003.

[11] H. Ding and I. Sølvberg, "Choosing appropriate peer-to-peer infrastructure for your digital libraries.," in *ICADL* (E. A. Fox, E. J. Neuhold, P. Premsmit, and V. Wuwongse, eds.), vol. 3815 of *Lecture Notes in Computer Science*, pp. 457–462, Springer, 2005.

[12] "MARC to Dublin Core crosswalk." http://www.loc.gov/marc/marc2dc.html, February, 2001. Network Development and MARC Standards Office, Library of Congress.

[13] B. M. Leiner, "The ncstrl approach to open architecture for the confederated digital library," *D-Lib Magazine*, December 1998.

[14] M. A. Gonçalves, R. K. France, and E. A. Fox, "Marian: Flexible interoperability for federated digital libraries.," in *ECDL*, pp. 173–186, 2001. http://link.springer.de/link/service/series/0558/bibs/2163/21630173.htm%.

[15] W. Y. Arms, *Digital Libraries.* The MIT Press, 2000.

[16] A. M. Ouksel and A. P. Sheth, "Semantic interoperability in global information systems: A brief introduction to the research area and the special section.," *SIGMOD Record*, vol. 28, no. 1, pp. 5–12, 1999.

[17] G. Koutrika, "Heterogeneity in Digital Libraries: Two Sides of the Same Coin." http://www.delos.info/newsletter/issue3/feature2/, Available: 2005.08.

[18] M. Lenzerini, "Data integration: A theoretical perspective," in *Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems(PODS)*, (Madison, Wisconsin, USA), pp. 233–246, 2002.

[19] A. Y. Levy, A. Rajaraman, and J. J. Ordille, "Querying heterogeneous information sources using source descriptions," in *VLDB*, pp. 251–262, 1996.

[20] A. Y. Levy, "Combibining artificial intelligence and databases for data integration," in *Artificial Intelligence Today*, pp. 249–268, Springer, 1999.

[21] A. Y. Halevy, "Learning about data integration challenges from day one.," *SIGMOD Record*, vol. 32, no. 3, pp. 16–17, 2003.

[22] S. R. J. Leonidas Galanis, Yuan Wang and D. J. DeWitt, "Locating data sources in large distributed systems," in *the 29th VLDB Conference*, (Berlin, Germany), 2003.

[23] D. K. Ioana Manolescu, Daniela Florescu, "Answering xml queries on heterogeneous data sources," in *VLDB*, pp. 241–250, 2001.

[24] M. A. Chaplan, "Mapping laborline thesaurus terms to library of congress subject headings: Implications for vocabulary switching," *Library Quarterly*, vol. 65, no. 1, 1995.

[25] J. A. H. Tim Berners-Lee and O. Lassila, "The semantic web,," *Scientific American,*, vol. 284(5), pp. 34–43, 2001.

[26] T. R. Gruber, "What is ontology?." http://www-ksl.stanford.edu/kst/what-is-an-ontology.html, 1993.

[27] S. Staab and et.al, "Trends and controversies: Where are the rules?," *IEEE Intelligent Systems*, vol. September/October, pp. 76–83, 2003.

[28] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML." http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/, May 2004.

[29] C. Lagoze and J. Hunter, "The abc ontology and model," in *Dublin Core Conference 2001*, (Tokyo, Japan), pp. 160–176, 2001.

[30] P. H. Jørgensen, "Cataloguing with xml, rdf, and ifla frbr," in *Proceedings of the 11th Nordic Conference on Information and Documentation*, 2001. http://www.bokis.is/iod2001/.

[31] "The CIDOC Conceptual Reference Model (CRM), the CIDOC CRM special interest group." http://cidoc.ics.forth.gr/, Available:2005.03.

[32] A. Paepcke, K. C.-C. Chang, H. Garcia-Molina, and T. Winograd, "Interoperability for digital libraries worldwide.," *Communication of the ACM*, vol. 41, no. 4, pp. 33–43, 1998.

[33] B. Ahlborn, W. Nejdl, and W. Siberski, "Oai-p2p: A peer-to-peer network for open archives.," in *ICPP Workshops*, pp. 462–468, IEEE Computer Society, 2002.

[34] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch, "EDUTELLA: a P2P networking infrastructure based on RDF," in *International World Wide Web Conferences (WWW)*, pp. 604–615, 2002.

[35] P. Haase, J. Broekstra, M. Ehrig, M. Menken, P. Mika, M. Olko, M. Plechawski, P. Pyszlak, B. Schnizler, R. Siebes, S. Staab, and C. Tempich, "Bibster - a semantics-based bibliographic peer-to-peer system.," in McIlraith *et al.* [249], pp. 122–136.

[36] B. M. Leiner, "The scope of the digital library," in *DLib Working Group on Digital Library Metrics*, 1998. http://www.dlib.org/metrics/public/papers/dig-lib-scope.html.

[37] D. J. Waters, "What are digital libraries?." http://www.clir.org/pubs/issues/issues04.html, 1998.

[38] C. L. Borgman, "What are digital libraries? competing visions.," *Information Processing & Management*, vol. 35, no. 3, pp. 227–243, 1999.

[39] M. Agosti, L. Candela, D. Castelli, N. Ferro, Y. Ioannidis, G. Koutrika, C. Meghini, P. Pagano, S. Ross, H.-J. Schek, and H. Schuldt, "Deliverable d1.4.2 a reference model for dlmss interim report (deliverable)." http://146.48.87.21:80/OLP/UI/1.0/Disseminate/1159911465iONUcJ37Hh/a221%159911465hTmzNhod, 2006.

[40] "RTF3650: Handle System Overview." http://www.ietf.org/rfc/rfc3650.txt, 2003.

[41] M. Buckland, *Redesigning Library Services: A Manifesto*. Chicago: ALA Books, 1992.

[42] C. L. Borgman, *From Gutenberg to the Global Information Infrastructure*. MIT Press, 2000.

[43] J. Hunter and S. Choudhury, "Working Towards MetaUtopia - A Survey of Current Metadata Research," *Library Trends, Organizing the Internet*, vol. 52(2), Fall, 2003. http://archive.dstc.edu.au/RDU/staff/jane-hunter/LibTrends_paper.pdf.

[44] W. Cathro, "Metadata: An overview." http://www.nla.gov.au/nla/staffpaper/cathro3.html, 1997. Seminar - "Matching Discovery and Recovery".

[45] C. Arms, "Some observations on metadata and digital libraries," in *Proceedings of the Bicentennial Conference on Bibliographic Control for the New Millennium*, 2001.

[46] O. Madison, J. John Byrum, S. Jouguelet, D. McGarry, N. Williamson, and M. Witt, "Functional requirements for bibliographic records," tech. rep., IFLA Universal Bibliographic Control and International MARC Programme, Munich, Germany, 1998. http://www.ifla.org/VII/s13/frbr/.pdf.

[47] "Digital libraries: Future directions for an european research programme," 2001. http://delos-noe.iei.pi.cnr.it/activities/researchforum/Brainstorming/b%rainstorming-report.pdf.

[48] R. Guenther and J. Radebaugh, "Understanding metadata." National Information Standard Organization (NISO) Press, Bethesda, USA, 2004. http://www.niso.org/standards/resources/UnderstandingMetadata.pdf.

[49] "The ACM Computing Classification System (1998) ." http://www.acm.org/class/1998/TOP.html, 1998.

[50] T. Berners-Lee, "Metadata architecture." http://www.w3.org/DesignIssues/Metadata.html, January 1997.

[51] D. Bearman, E. Miller, G. Rust, J. Trant, and S. Weibel, "A common model to support interoperable metadata," *D-Lib Magazine*, vol. 5, no. 1, 1999.

[52] J. Blixrud and et.al., *International Standard Serial Numbering (ISSN)*. NISO Press, 1995. http://www.niso.org/standards/resources/Z39-9.pdf.

[53] F. Schwarz and C. Hepfer, "Changes to the serial item and contribution identifier and the effects of those on publishers and libraries," *The Serials Librarian*, vol. 28(3/4), 1996.

[54] N. Paskin, "Doi: Current status and outlook," *DLIb Magazine*, vol. 5, no. 5, 1999. http://www.dlib.org/dlib/may99/05paskin.html.

[55] "RTF4122: A Universally Unique IDentifier (UUID) URN Namespace." http://www.ietf.org/rfc/rfc4122.txt, 2005.

[56] "Naming and Addressing: URIs, URLs, ..." http://www.w3.org/Addressing, 2005.

[57] "RTF3986: Uniform Resource Identifier (URI): Generic Syntax." http://www.acm.org/class/1998/TOP.html, 2005.

[58] D. Connolly, "Untangle URIs, URLs, and URNs." http://www-128.ibm.com/developerworks/xml/library/x-urlni.html, 2005.

[59] "RTF2141: URN Syntax." http://www.ietf.org/rfc/rfc2141.txt, 1997.

[60] "The MARC 21 formats: Background and principles, library of congress." http://www.loc.gov/marc/96principl.html, 2006.

[61] "What is a MARC record, and why is it important." http://www.loc.gov/marc/umb/um01to06.html, 2006.

[62] "MARC 21 format for bibliographic data, 1999 english edition." http://www.loc.gov/marc/bibliographic/ecbdlist.html.

[63] S. L. Weibel and T. Koch, "The dublin core metadata initiative mission, current activities, and future directions," *D-Lib Magazine*, vol. 6, no. 12, 2000.

[64] T. Baker, "A grammar of dublin core," *D-Lib Magazine*, vol. Volume 6 Number 10, 2000. http://www.dlib.org/dlib/october00/baker/10baker.html.

[65] "Information Retrieval (Z39.50): Application Service Definition and Protocol Specification." http://www.loc.gov/z3950/agency/Z39-50-2003.pdf, 2003.

[66] C. Lagoze and H. V. de Sompel, "The open archives initiative protocol for metadata harvesting." http://www.openarchives.org/OAI/openarchivesprotocol.html, 2002.

[67] C. Lagoze and J. R. Davis, "Dienst: An architecture for distributed document libraries.," *Commun. ACM*, vol. 38, no. 4, p. 47, 1995.

[68] A. Paepcke, M. Q. W. Baldonado, K. C.-C. Chang, S. B. Cousins, and H. Garcia-Molina, "Using distributed objects to build the stanford digital library infobus.," *IEEE Computer*, vol. 32, no. 2, pp. 80–87, 1999.

[69] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. D. Ullman, and J. Widom, "The TSIMMIS project: Integration of heterogeneous information sources," *Journal of Intelligent Information System*, vol. 8(2), pp. 117–132, 1997.

[70] H. V. de Sompel and C. Lagoze, "The Santa Fe Convention of the Open Archives Initiative," *D-Lib Magazine*, vol. Volume 6, no. 2, 2000.

[71] R. Sanderson and et.al., "SRW: Search/Retrieve Webservice. Version: 1.1." http://srw.cheshire3.org/SRW-1.1.pdf, 2004.

[72] C. Lagoze and H. V. de Sompel, "The open archives initiative: building a low-barrier interoperability framework.," in *JCDL*, pp. 54–62, ACM, 2001.

[73] "The simple digital library interoperability protocol." http://dbpubs.stanford.edu:8091/ testbed/doc2/SDLIP/.

[74] OCLC, "Introduction to Dewey Decimal Classification." http://www.oclc.org/dewey/versions/ddc22print/intro.pdf, Available: 2005.08.

[75] K. S. Jones and P. Willett, *Readings in Information Retrieval*. Morgan Kaufmann Publishers, INC., 1997.

[76] D. L. McGuinness, *Ontologies Come of Age*, ch. Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential. Dieter Fensel and Jim Hendler and Henry Lieberman and Wolfgang Wahlster, MIT Press, 2002.

[77] J. Brase, W. Nejdl, M. Painter, M. Sintek, and U. Thaden, "Intelligently authoring metadata for a semantic web peer-to-peer environment." http://www.kbs.uni-hannover.de/Arbeiten/Publikationen/2003/ISWC_long.p%df, 2003.

[78] S. Little and J. Hunter, "Rules-by-example - a novel approach to semantic indexing and querying of images.," in McIlraith *et al.* [249], pp. 534–548.

[79] P. J. Hane, "The truth about federated searching," *Information Today Magazine*, vol. 20, Nov./Dec. 2003.

[80] R. R. Korfhage, *Information Storage and Retrieval*. John Wiley, 1997.

[81] W. Meng, Z. Wu, C. T. Yu, and Z. Li, "A highly scalable and effective method for metasearch.," *ACM Transaction of Information System*, vol. 19, no. 3, pp. 310–335, 2001.

[82] B. R. Schatz, W. H. Mischo, T. W. Cole, A. P. Bishop, S. Harum, E. H. Johnson, L. J. Neumann, H. Chen, and D. Ng, "Federated search of scientific literature.," *IEEE Computer*, vol. 32, no. 2, pp. 51–59, 1999.

[83] D. Brickley, J. Hunter, C. Lagoze, L. Miller, and W. Ren, "The harmony project home page." http://www.ilrt.bris.ac.uk/discovery/harmony/, 1999.

[84] J. Hunter, J. Drennan, and S. Little, "Realizing the hydrogen economy through semantic web technologies.," *IEEE Intelligent Systems*, vol. 19, no. 1, pp. 40–47, 2004.

[85] "DELOS Digital library architecture (WP1)- Cluster Objectives." http://www.delos.info/WP1.html, 2004.

[86] H. Ding, I. Sølvberg, and Y. Lin, "A vision on semantic retrieval in p2p network.," in *AINA (1)*, pp. 177–182, IEEE Computer Society, 2004.

[87] H. Ding and I. Sølvberg, "Towards the schema heterogeneity in distributed digital libraries," in *6th International Conference on Enterprise Information System (ICEIS)*, vol. 5, (Porto, Portugal), April 2004.

[88] G. Koutrika, "Newsletter: Heterogeneity in Digital Libraries: Two Sides of the Same Coin." http://www.delos.info/newsletter/issue3/feature2/, 2005.

[89] S. E. Madnick, "Are we moving toward an information superhighway or a tower of babel? the challenge of large-scale semantic heterogeneity.," in *ICDE* (S. Y. W. Su, ed.), pp. 2–8, IEEE Computer Society, 1996.

[90] M. Doerr, "Semantic problems of thesaurus mapping," *Journal of Digital Information*, vol. Volume 1 Issue 8, 2001.

[91] E. McCulloch, A. Shiri, and D. Nicholson, "Challenges and issues in terminology mapping: a digital library perspective.," *The Electronic Library, Emerald*, vol. 23, no. 6, pp. 671–677, 2005.

[92] Ó. Corcho and A. Gómez-Pérez, "A layered model for building ontology translation systems.," *International Journal of Semantic Web Information System*, vol. 1, no. 2, pp. 22–48, 2005.

[93] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.

[94] D. Dou, D. V. McDermott, and P. Qi, "Ontology translation on the semantic web.," *Journal of Data Semantics*, vol. 2, pp. 35–57, 2005.

[95] M. Doerr, J. Hunter, and C. Lagoze, "Towards a core ontology for information integration.," *J. Digit. Inf.*, vol. 4, no. 1, 2003.

[96] J. R. Davis and C. Lagoze, "Ncstrl: Design and deployment of a globally distributed digital library.," *JASIS*, vol. 51, no. 3, pp. 273–280, 2000.

[97] D. Gourley, "Managing change: An architecture for the evolving digital library," in *EDUCAUSE Annual Conference*, (Indiana, USA), October 2831, 2001.

[98] I. T. Foster, "The anatomy of the grid: Enabling scalable virtual organizations.," in *Euro-Par* (R. Sakellariou, J. Keane, J. R. Gurd, and L. Freeman, eds.), vol. 2150 of *Lecture Notes in Computer Science*, pp. 1–4, Springer, 2001.

[99] M. P. Papazoglou, "Service -oriented computing: Concepts, characteristics and directions," in *Fourth International Conference on Web Information Systems Engineering (WISE'03)*, December 2003.

[100] H. Ding and I. Sølvberg, "Exploiting extended service-oriented architecture for federated digital libraries.," in *ICADL*, pp. 184–194, 2004.

[101] Lime Wire LLC, "Gnutella - limewire 4.0." http://www.limewire.com/, 2000.

[102] A. I. T. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems.," in *Middleware* (R. Guerraoui, ed.), vol. 2218 of *Lecture Notes in Computer Science*, pp. 329–350, Springer, 2001.

[103] A. Crespo and H. Garcia-Molina, "Routing indices for peer-to-peer systems," in *Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)*, 2002.

[104] A. Crespo and H. Garcia-Molina, "Semantic overlay networks for p2p systems," tech. rep., Computer Science Department, Stanford University, 2002.

[105] M. H. Karl Aberer, Philippe Cudre-Mauroux, "The chatty web: Emergent semantics through gossiping," in *The International World Wide Web Conference*, (Hungary), 2003.

[106] L. Gong, "Industry report: Jxta: A network programming environment," *Internet Computing, IEEE*, vol. 5(3), May-June 2001.

[107] "KaZaa - a completely distributed peer-to-peer file sharing service." http://www.kazaa.com/, Available: 2005.03.

[108] "Napster." http://www.napster.com, 2001.

[109] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," in *International Workshop on Design Issues in Anonymity and Unobservability, LNCS 2009*, vol. 2009 of *Lecture Notes in Computer Science*, (Berkeley, CA, USA), pp. 46–66, Springer, 2000.

[110] S. Jain, R. Mahajan, and D. Wetherall, "A Study of the Performance Potential of DHT-based Overlays," in *USENIX Symposium on Internet Technologies and Systems 2003*, (Seattle, Washington, USA), 2003.

[111] D. K. Ion Stoica, Robert Morris, F. Kaashoek, and H. Balakrishnan, "Chold: A scalable peer-to-peer lookup service for internet applications," in *SIGCOMM*, (San Diego, California, USA), 2001.

[112] S. Ratnasamy, P. Francis, M. Handley, R. Karp1, and S. Shenker, "A scalable content-addressable network," in *SIGCOMM*, (San Diego, California, USA), 2001.

[113] C. Tang, Z. Xu, and M. Mahalingam, "psearch: Information retrieval in structured overlays," in *Proceedings of HotNetsI02*, (Princeton, New Jersey, USA), 2002.

[114] Z. Xu, C. Tang, and Z. Zhang, "Building topology-aware overlays using global soft-state," in *Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS'03)*, 2002.

[115] K. Aberer, P. Cudré-Mauroux, A. Datta, Z. Despotovic, M. Hauswirth, M. Punceva, and R. Schmidt, "P-grid: a self-organizing structured p2p system.," *SIGMOD Record*, vol. 32, no. 3, pp. 29–33, 2003.

[116] M. T. Schlosser, M. Sintek, S. Decker, and W. Nejdl, "Hypercup - hypercubes, ontologies, and efficient search on peer-to-peer networks.," in *AP2PC*, vol. 2530 of *Lecture Notes in Computer Science*, pp. 112–124, Springer, 2002.

[117] A. Y. Halevy, Z. G. Ives, P. Mork, and I. Tatarinov, "Piazza: data management infrastructure for semantic web applications.," in *WWW*, pp. 556–567, 2003.

[118] S. Oaks, B. Traversat, and L. Gong, *JXTA in a Nutshell*. O'Reilly & Associates, Inc., September, 2002.

[119] M. Cai and M. Frank, "Rdfpeers: A scalable distributed rdf repository based on a structured peer-to-peer network," in *The 13th International World Wide Web Conference (WWW)*, pp. 17–22, 2004.

[120] D. Liben-Nowell, H. Balakrishnan, and D. R. Karger, "Analysis of the evolution of peer-to-peer systems.," in *PODC*, pp. 233–242, 2002.

[121] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz., "Handling churn in a dht," in *Proceedings of the USENIX Annual Technical Conference*, (Boston, USA), 2004.

[122] B. Yang and H. Garcia-Molina, "Designing a super-peer network," in *IEEE International Conference on Data Engineering, 2003*, 2003. http://dbpubs.stanford.edu:8090/pub/showDoc.Fulltext?lang=en\&doc=2003-%33\&format=pdf\&compression=.

[123] S. Amer-Yahia and J. Shanmugasundaram, "Xml full-text search: Challenges and opportunities.," in *VLDB* (K. Böhm, C. S. Jensen, L. M. Haas, M. L. Kersten, P.-Å. Larson, and B. C. Ooi, eds.), p. 1368, ACM, 2005.

[124] W.-T. Balke, W. Nejdl, W. Siberski, and U. Thaden, "Dl meets p2p - distributed document retrieval based on classification and content.," in *ECDL* (A. Rauber, S. Christodoulakis, and A. M. Tjoa, eds.), vol. 3652 of *Lecture Notes in Computer Science*, pp. 379–390, Springer, 2005.

[125] H. T. Shen, Y. Shu, and B. Yu, "Efficient semantic-based content search in p2p network.," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 7, pp. 813–826, 2004.

[126] e. Greg Karvounarakis, "The RDF Query Language (RQL)." http://139.91.183.30:9090/RDF/RQL/, 2003.

[127] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip algorithms: Design, analysis and applications," in *Proc. IEEE Infocom 2005, Volume 3*, (Miami, USA), pp. 1653–1664, 2005.

[128] C. R. Palmer and J. G. Steffan, "Generating network topologies that obey power laws," in *Proceedings of GLOBECOM '2000*, November 2000.

[129] Paul Miller, "UK Interoperability Focus." http://www.ukoln.ac.uk/interop-focus/about/, 2000.

[130] M. Friedman, A. Y. Levy, and T. D. Millstein, "Navigational plans for data integration.," in *AAAI/IAAI*, pp. 67–73, 1999.

[131] D. Calvanese, E. Damaggio, G. D. Giacomo, M. Lenzerini, and R. Rosati, "Semantic data integration in p2p systems.," in *DBISP2P*, pp. 77–90, 2003. http://springerlink.metapress.com/openurl.asp?genre=article{\&}issn=030%2-9743{\&}volume=2944{\&}spage=77.

[132] D. Calvanese, G. D. Giacomo, M. Lenzerini, and R. Rosati, "Logical foundations of peer-to-peer data integration.," in *PODS*, pp. 241–251, 2004. http://www.acm.org/sigmod/pods/proc04/pdf/P-25.pdf.

[133] G. H. Leazer and R. P. Smiraglia, "Toward the bibliographic control of works: Derivative bibliographic relationships in an online union catalog," in *Proceedings of the 1st ACM International Conference on Digital Libraries, March 20-23, 1996, Bethesda, Maryland, USA*, pp. 36–43, ACM, 1996.

[134] C. J. Godby, J. A. Young, and E. Childress, "A repository of metadata crosswalks," *D-Lib Magazine*, vol. 10, no. 12, 2004.

[135] M. L. Zeng, "Supporting metadata interoperability: Trends and issues.," in *Global Digital Library Development in the New Millennium.* (C.-C. Chen, ed.), pp. 405–412, 2001.

[136] J. Hunter and C. Lagoze, "Combining rdf and xml schemas to enhance interoperability between metadata application profiles.," in *WWW*, pp. 457–466, 2001.

[137] "TV-Anytime forum." http://www.tv-anytime.org/.

[138] "MPEG-21 multimedia framework." http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm.

[139] "The BIBLINK core application profile." http://www.schemas-forum.org/registry/schemas/biblink/BC-schema.html.

[140] G. Rust and M. Bide, "The indecs metadata schema building blocks." http://www.indecs.org/pdf/model1.pdf, 1999.

[141] "MPEG-7 overview." http://www.chiariglione.org/MPEG/standards/mpeg-7/mpeg-7.htm.

[142] "Content Standard for Digital Geospatial Metadata (CSDGM)." http://www.fgdc.gov/standards/projects/FGDC-standards-projects/metadata%/base-metadata/v2_0698.pdf.

[143] "The gateway to educational materials." http://www.thegateway.org.

[144] "Draft Standard for Learning Object Metadata(LTSC-LOM)." http://ltsc.ieee.org/wg12/, 2002.

[145] R. Heery and M. Patel, "Application profiles: mixing and matching metadata schemas," *Ariadne Magazine*, vol. 25, September, 2000.

[146] D. Brickley, R. Guha, and B. McBride, "RDF Vocabulary Description Language 1.0: RDF Schema." http://www.w3.org/TR/rdf-schema/, February 2004.

[147] D. C. Fallside, H. S. Thompson, D. Beech, M. Maloney, N. Mendelsohn, and etc, "W3c architecture domain - xml schema." http://www.w3.org/XML/Schema, May 2001.

[148] "The SCHEMAS Project." http://www.schemas-forum.org/. Forum for Metadata Schema Implementers.

[149] T. Baker, M. Dekkers, R. Heery, M. Patel, and G. Salokhe, "What terms does your metadata use? application profiles as machine-understandable narratives.," *Journal of Digital Information*, vol. 2, no. 2, 2001.

[150] "CORES project." http://www.cores-eu.net/, Available: 2005.08.

[151] "DCMI Registry." http://dublincore.org/groups/registry/, Available: 2005.08.

[152] L. M. Chan, "Metadata Interoperability: A Study of Methodology," *Chinese Librarianship: an International Electronic Journal*, vol. No.19 (June), 2005.

[153] F. Manola and E. Miller, "Rdf primer." http://www.w3.org/TR/rdf-primer/, February 2004.

[154] D. L. McGuinness and F. van Harmelen, "Owl web ontology language overview." http://www.w3.org/TR/owl-features/, February 2004.

[155] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Third Edition)." http://www.w3.org/TR/REC-xml/, Available: 2005.08.

[156] S. Mazzocchi, "Simile: Objectives, current status, and demonstration," in *International CIDOC CRM Workshop*, (Heraklion, Crete, Greece), 2004.

[157] T. Bray, "RDF and Metadata." http://www.xml.com/pub/a/98/06/rdf.html, June 09,1998.

[158] "METS Overview & Tutorial." http://www.loc.gov/standards/mets/METSOverview.v2.html, May 24, 2005.

[159] "Metadata Encoding and Transmission Standard (METS)." http://www.loc.gov/standards/mets/, Available: 2005.08.

[160] R. Tenant, "Digital library - different paths to interoperability," *Library Journal*, vol. 126(3), pp. 118–119, 2001. http://www.libraryjournal.com/article/CA156525.html.

[161] T. Baker, R. Clayphan, and P. Johnston, "Tutorial: Creating an application profile." http://dublincore.org/resources/training/dc-2004/english/DC-2004_Tutor%ial_3_1_en.pdf, 2004.

[162] D. Fensel, F. van Harmelen, I. Horrocks, D. L. McGuinness, and P. F. Patel-Schneider, "Oil: An ontology infrastructure for the semantic web.," *IEEE Intelligent Systems*, vol. 16, no. 2, pp. 38–45, 2001.

[163] R. Engels and B. Bremdal, "Information extraction: State-of-the-art report," tech. rep., IST Project: On-To-Knowledge, 2000.

[164] G. Gardarin, H. Kou, K. Zeitouni, X. Meng, and H. Wang, "Sewise: An ontology-based web information search engine.," in *NLDB* (A. Düsterhöft and B. Thalheim, eds.), vol. 29 of *LNI*, pp. 106–119, GI, 2003.

[165] A. P. Sheth and C. Ramakrishnan, "Semantic (web) technology in action: Ontology driven information systems for search, integration and analysis.," *IEEE Data Engineering Bulletin*, vol. 26, no. 4, pp. 40–48, 2003.

[166] L. Zhang, Y. Yu, J. Zhou, C. Lin, and Y. Yang, "An enhanced model for searching in semantic portals.," in *WWW* (A. Ellis and T. Hagino, eds.), pp. 453–462, ACM, 2005.

[167] P. Warren, "Applying semantic technologies to a digital library: a case study," *Library Management Journal, Emerald*, vol. 26, no. 4/5, pp. 196–205, 2005.

[168] Y. Sure and R. Studer, "Semantic web technologies for digital libraries," *Library Management Journal, Emerald*, vol. 26, no. 4/5, pp. 190–195, 2005.

[169] C. A. Lynch and H. Garcia-Molina, "Interoperability, scaling, and the digital libraries research agenda," in *IITA Digital Libraries Workshop*, August 1995.

[170] "Metadata Object Description Schema(MODS)." http://www.loc.gov/standards/mods/, Available at:2005.03.

[171] T. R. Gruber, "Toward principles for the design of ontologies: Used for knowledge sharing," in *Formal Ontology in Conceptual Analysis and Knowledge Representation* (N. Guarino, ed.), (Deventer, The Netherlands), Kluwer Academic Publishers, 1993. Available as Technical Report KSL 93-04, Knowledge Systems Laboratory, Stanford University. http://www-ksl.stanford.edu/knowledge-sharing/papers/onto-design.ps.

[172] B. Chandrasekaran, J. R. Josephson, and R. Benjamins, "What are ontologies, and why do we need them?," *IEEE Intelligent System*, vol. January/February, 1999.

[173] Lars Marius Garshol , "Metadata? Thesauri? Taxonomies? Topic Maps!." http://www.ontopia.net/topicmaps/materials/tm-vs-thesauri.html, 2004.

[174] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen, "From shiq and rdf to owl: the making of a web ontology language.," *Journal of Web Semantics*, vol. 1, no. 1, pp. 7–26, 2003.

[175] T. Berners-Lee, *Weaving the Web*. San Francisco, USA: Harper, 1997.

[176] G. Klyne and J. J. Carroll, "Resource description framework (rdf):concepts and abstract syntax." http://www.w3.org/TR/rdf-concepts/, February 2004.

[177] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen, "Reviewing the design of daml+oil: An ontology language for the semantic web.," in *AAAI/IAAI*, pp. 792–797, 2002.

[178] J. Z. Pan and I. Horrocks, "Rdfs(fa) and rdf mt: Two semantics for rdfs.," in Fensel *et al.* [250], pp. 30–46.

[179] R. J. Brachman and H. J. Levesque, "The tractability of subsumption in frame-based description languages.," in *AAAI*, pp. 34–37, 1984.

[180] T. Catarci and M. Lenzerini, "Representing and using interschema knowledge in cooperative information systems.," *International Journal of Cooperative Information Systems (IJCIS)*, vol. 2, no. 4, pp. 375–398, 1993.

[181] Y. Arens, C. A. Knoblock, and W.-M. Shen, "Query reformulation for dynamic information integration.," *Journal of Intelligent Information Systems (JIIS)*, vol. 6, no. 2/3, pp. 99–130, 1996.

[182] D. Calvanese, G. D. Giacomo, M. Lenzerini, D. Nardi, and R. Rosati, "Information integration: Conceptual modeling and reasoning support.," in *CoopIS*, pp. 280–291, IEEE Computer Society, 1998.

[183] E. Mena, V. Kashyap, A. Illarramendi, and A. P. Sheth, "Imprecise answers in distributed environments: Estimation of information loss for multi-ontology based query processing.," *International Journal of Cooperative Information Systems (IJCIS)*, vol. 9, no. 4, pp. 403–425, 2000.

[184] S. Bergamaschi, S. Castano, M. Vincini, and D. Beneventano, "Semantic integration of heterogeneous information sources.," *Data Knowl. Eng.*, vol. 36, no. 3, pp. 215–249, 2001.

[185] F. Baader, I. Horrocks, and U. Sattler, "Description logics," in Staab and Studer [251], pp. 3–28.

[186] A. P. Sheth, "From semantic search & integration to analytics.," in Kalfoglou *et al.* [252].

[187] D. Calvanese, G. D. Giacomo, M. Lenzerini, and D. Nardi, "Reasoning in expressive description logics," in *Handbook of Automated Reasoning* (J. A. Robinson and A. Voronkov, eds.), pp. 1581–1634, Elsevier and MIT Press, 2001.

[188] I. Horrocks, "Description logics in ontology applications.," in *KI* (U. Furbach, ed.), vol. 3698 of *Lecture Notes in Computer Science*, p. 16, Springer, 2005.

[189] I. Horrocks, U. Sattler, and S. Tobies, "Reasoning with individuals for the description logic shiq.," in *CADE* (D. A. McAllester, ed.), vol. 1831 of *Lecture Notes in Computer Science*, pp. 482–496, Springer, 2000.

[190] M. A. Musen and S. W. Tu, "Problem-solving models for generation of task-specific knowledge-acquisition tools.," in *AIFIPP* (J. Cuena, ed.), vol. A-27 of *IFIP Transactions*, pp. 23–49, North-Holland, 1992.

[191] S. W. Tu, H. Eriksson, J. H. Gennari, Y. Shahar, and M. A. Musen, "Ontology-based configuration of problem-solving methods and generation of knowledge-acquisition tools: application of protege-ii to protocol-based decision support.," *Artificial Intelligence in Medicine*, vol. 7, no. 3, pp. 257–289, 1995.

[192] I. Horrocks, B. Parsia, P. F. Patel-Schneider, and J. A. Hendler, "Semantic web architecture: Stack or two towers?.," in *PPSWR* (F. Fages and S. Soliman, eds.), vol. 3703 of *Lecture Notes in Computer Science*, pp. 37–41, Springer, 2005.

[193] C. Golbreich, O. Dameron, B. Gibaud, and A. Burgun, "Web ontology language requirements w.r.t expressiveness of taxonomy and axioms in medicine.," in Fensel *et al.* [250], pp. 180–194.

[194] T. K. Huwe, "Keep those web skills current," *Computers in Libraries*, vol. 24, no. 8, 2004.

[195] J. F. Sowa, "Ontology, metadata, and semiotics.," in *ICCS* (B. Ganter and G. W. Mineau, eds.), vol. 1867 of *Lecture Notes in Computer Science*, pp. 55–81, Springer, 2000.

[196] M. Uschold and M. Grüninger, "Ontologies: principles, methods, and applications," *Knowledge Engineering Review*, vol. 11, no. 2, pp. 93–155, 1996.

[197] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller, "Introduction to wordnet: An on-line lexical database," *Cognitive Science Laboratory, Princeton University*, August 1993.

[198] S. Staab, R. Studer, H.-P. Schnurr, and Y. Sure, "Knowledge processes and ontologies," *IEEE Intelligent Systems*, vol. 16, no. 1, pp. 26–34, 2001.

[199] Y. Sure, S. Staab, and R. Studer, "On-to-knowledge methodology (otkm).," in Staab and Studer [251], pp. 117–132.

[200] J. A. Hendler, "Agents and the semantic web," *IEEE Intelligent Systems*, vol. 16(2), pp. 30–37, 2001.

[201] B. Ganter and R. Wille, *Formal Concept Analysis — Mathematical Foundations*. New York: Spinger-Verlag, 1997.

[202] O. Corcho, M. Fernández-López, A. Gómez-Pérez, and et.al., "Deliverable 1.3: A survey on ontology tools," tech. rep., OntoWeb, 2002. http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/OntoWeb_Del_1-3%.pdf.

[203] C. C. Marshall, "Toward an ecology of hypertext annotation.," in *Hypertext*, pp. 40–49, ACM, 1998.

[204] S. Bechhofer, L. Carr, C. A. Goble, S. Kampa, and T. Miles-Board, "The semantics of semantic annotation.," in *CoopIS/DOA/ODBASE* (R. Meersman and Z. Tari, eds.), vol. 2519 of *Lecture Notes in Computer Science*, pp. 1152–1167, Springer, 2002.

[205] "Swiss-Prot Protein Knowledgebase ." http://ca.expasy.org/sprot/.

[206] D. Fensel, S. Decker, M. Erdmann, and R. Studer, "Ontobroker in a nutshell.," in *ECDL* (C. Nikolaou and C. Stephanidis, eds.), vol. 1513 of *Lecture Notes in Computer Science*, pp. 663–664, Springer, 1998.

[207] J. Heflin, J. A. Hendler, and S. Luke, "Shoe: A blueprint for the semantic web.," in *Spinning the Semantic Web* (D. Fensel, J. A. Hendler, H. Lieberman, and W. Wahlster, eds.), pp. 29–63, MIT Press, 2003.

[208] L. Carr, W. Hall, S. Bechhofer, and C. A. Goble, "Conceptual linking: ontology-based open hypermedia.," in *WWW*, pp. 334–342, 2001.

[209] L. Reeve and H. Han, "Survey of semantic annotation platforms.," in *SAC* (H. Haddad, L. M. Liebrock, A. Omicini, and R. L. Wainwright, eds.), pp. 1634–1638, ACM, 2005.

[210] A. Maedche and S. Staab, "Ontology learning for the semantic web.," *IEEE Intelligent Systems*, vol. 16, no. 2, pp. 72–79, 2001.

[211] M. H. Karl Aberer, Philippe Cudre-Mauroux, "Combining and relating ontologies: an analysis of problems and solutions," in *IJCAI'01 Workshop on Ontologies and Information Sharing*, (Seattle, USA), 2001.

[212] H. Wache, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and et.al, "Ontology-based integration of information - a survey of existing approaches," in *IJCAI-01 Workshop: Ontologies and Information Sharing*, 2001.

[213] D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder, "An environment for merging and testing large ontologies.," in *KR*, pp. 483–493, 2000.

[214] Y. Kalfoglou and M. Schorlemmer, "Ontology mapping: The state of the art," in *Semantic Interoperability and Integration* (Y. Kalfoglou, M. Schorlemmer, A. Sheth, S. Staab, and M. Uschold, eds.), no. 04391 in Dagstuhl Seminar Proceedings, Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany, 2005. http://drops.dagstuhl.de/opus/volltexte/2005/40/pdf/04391.KalfoglouYann%is.Paper.40.pdf.

[215] G. Stumme and A. Maedche, "Ontology merging for federated ontologies on the semantic web," in *IJCAI-01 Workshop: Ontologies and Information Sharing*, 2001.

[216] H. Stuckenschmidt and F. van Harmelen, "Ontology-based metadata generation from semi-structured information.," in *K-CAP*, pp. 163–170, ACM, 2001.

[217] J. F. Sowa, "Building, sharing, and merging ontologies." http://www.jfsowa.com/ontology/ontoshar.htm.

[218] Y. Kalfoglou and W. M. Schorlemmer, "Ontology mapping: The state of the art.," in Kalfoglou *et al.* [252].

[219] F. van Harmelen, "Ontology mapping: A way out of the medical tower of babel?," in *AIME* (S. Miksch, J. Hunter, and E. T. Keravnou, eds.), vol. 3581 of *Lecture Notes in Computer Science*, pp. 3–6, Springer, 2005.

[220] P. Bouquet, L. Serafini, and S. Zanobini, "Semantic coordination: A new approach and an application.," in Fensel *et al.* [250], pp. 130–145.

[221] X. Su, *Semantic Enrichment for Ontology Mapping.* PhD thesis, Norwegian University of Science and Technology, October 2004.

[222] M. Ehrig, S. Staab, and Y. Sure, "Bootstrapping ontology alignment methods with apfel," in *International Semantic Web Conference* (Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, eds.), vol. 3729 of *Lecture Notes in Computer Science*, pp. 186–200, Springer, 2005.

[223] H. Chalupsky, "Ontomorph: A translation system for symbolic knowledge.," in *KR*, pp. 471–482, 2000.

[224] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy.," in *IJCAI*, pp. 448–453, 1995.

[225] A. Borgida and L. Serafini, "Distributed description logics: Assimilating information from peer sources.," *Journal of Data Semantics*, vol. 1, pp. 153–184, 2003.

[226] P. Mitra, G. Wiederhold, and M. L. Kersten, "A graph-oriented model for articulation of ontology interdependencies," in *EDBT* (C. Zaniolo, P. C. Lockemann, M. H. Scholl, and T. Grust, eds.), vol. 1777 of *Lecture Notes in Computer Science*, pp. 86–100, Springer, 2000.

[227] C. Ghidini and F. Giunchiglia, "Local models semantics, or contextual reasoning=locality+compatibility.," *Artif. Intell.*, vol. 127, no. 2, pp. 221–259, 2001.

[228] B. Xu, P. Wang, J. Lu, Y. Li, and D. Kang, "Theory and semantic refinement of bridge ontology based on multi-ontologies.," in *ICTAI*, pp. 442–449, IEEE Computer Society, 2004.

[229] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt, "Contextualizing ontologies.," *Journal of Web Semantics*, vol. 1, no. 4, pp. 325–343, 2004.

[230] C. Golbreich, "Combining rule and ontology reasoners for the semantic web.," in *RuleML* (G. Antoniou and H. Boley, eds.), vol. 3323 of *Lecture Notes in Computer Science*, pp. 6–22, Springer, 2004.

[231] H. Boley, B. Grosof, M. Sintek, S. Tabet, and G. Wagner, "RuleML Design." http://www.ruleml.org/indesign.html, 2002.

[232] I. Horrocks and P. F. Patel-Schneider, "A proposal for an owl rules language.," in *WWW* (S. I. Feldman, M. Uretsky, M. Najork, and C. E. Wills, eds.), pp. 723–731, ACM, 2004.

[233] B. N. Grosof, I. Horrocks, R. Volz, and S. Decker, "Description logic programs: combining logic programs with description logic.," in *WWW*, pp. 48–57, 2003.

[234] U. Hustadt, B. Motik, and U. Sattler, "Reducing shiq-description logic to disjunctive datalog programs.," in *KR* (D. Dubois, C. A. Welty, and M.-A. Williams, eds.), pp. 152–162, AAAI Press, 2004.

[235] I. Horrocks, U. Sattler, and S. Tobies, "Practical reasoning for very expressive description logics," *Logic Journal of the IGPL*, vol. 8, no. 3, 2000.

[236] B. Motik, U. Sattler, and R. Studer, "Query answering for owl-dl with rules.," in McIlraith *et al.* [249], pp. 549–563.

[237] I. Horrocks, "Applications of description logics: State of the art and research challenges.," in *ICCS* (F. Dau, M.-L. Mugnier, and G. Stumme, eds.), vol. 3596 of *Lecture Notes in Computer Science*, (Kassel, Germany), pp. 78–90, Springer, 2005.

[238] "Hamming distance." http://en.wikipedia.org/wiki/Hamming_distance, 2006.

[239] "Levenshtein distance." http://en.wikipedia.org/wiki/Levenshtein_distance, 2000.

[240] A. Doan, J. Madhavan, P. Domingos, and A. Y. Halevy, "Learning to map between ontologies on the Semantic Web.," in *WWW*, pp. 662–673, 2002.

[241] G. Salton and M. McGill, *Introduction to Modern Information Retrieval*. New York, NY: McGraw-Hill, 1983.

[242] C. van Rijsbergen, *Information Retrieval*. London: Butterworths, second ed., 1979.

[243] "Owlapi." http://sourceforge.net/projects/owlapi.

[244] "Kaon2." http://kaon2.semanticweb.org/.

[245] C. Y. Kong, C.-L. Wang, and F. C. M. Lau, "Ontology mapping in pervasive computing environment.," in *EUC* (L. T. Yang, M. Guo, G. R. Gao, and N. K. Jha, eds.), vol. 3207 of *Lecture Notes in Computer Science*, pp. 1014–1023, Springer, 2004.

[246] H. Lab, "Jena - a semantic web framework for java." http://jena.sourceforge.net/, 2004.

[247] H. Ding, "Towards the metadata integration issues in peer-to-peer based digital libraries.," in *GCC* (H. Jin, Y. Pan, N. Xiao, and J. Sun, eds.), vol. 3251 of *Lecture Notes in Computer Science*, pp. 851–854, Springer, 2004.

[248] "SPARQL Query Language for RDF." http://www.w3.org/TR/rdf-sparql-query/.

[249] S. A. McIlraith, D. Plexousakis, and F. van Harmelen, eds., *The Semantic Web - ISWC 2004: Third International Semantic Web Conference,Hiroshima, Japan, November 7-11, 2004. Proceedings*, vol. 3298 of *Lecture Notes in Computer Science*, Springer, 2004.

[250] D. Fensel, K. P. Sycara, and J. Mylopoulos, eds., *The Semantic Web - ISWC 2003, Second International Semantic Web Conference, Sanibel Island, FL, USA, October 20-23, 2003, Proceedings*, vol. 2870 of *Lecture Notes in Computer Science*, Springer, 2003.

[251] S. Staab and R. Studer, eds., *Handbook on Ontologies*. International Handbooks on Information Systems, Springer, 2004.

[252] Y. Kalfoglou, W. M. Schorlemmer, A. P. Sheth, S. Staab, and M. Uschold, eds., *Semantic Interoperability and Integration*, vol. 04391 of *Dagstuhl Seminar Proceedings*, IBFI, Schloss Dagstuhl, Germany, 2005.