AUTOMATIC SEMANTIC HEADER GENERATOR FOR PDF DOCUMENTS

FURONG XUE

A THESIS

IN

THE DEPARTMENT

OF

COMPUTER SCIENCE

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE
CONCORDIA UNIVERSITY
MONTREAL, QUEBEC, CANADA

DECEMBER 2003

Your file Votre référence
ISBN: 0-612-91140-3
Our file Notre référence
ISBN: 0-612-91140-3

Abstract

Automatic Semantic Header Generator for PDF Documents

Furong Xue

The Concordia INdexing and DIscovery system (CINDI) is an information discovery and retrieval system to enable a reader to discover resources from a bibliographic database. It uses a metadata description called semantic header to describe an information resource, whose content includes title, author name, the subject and sub-subject, etc. Automatic Semantic Header Generator (ASHG) is used to generate a draft version of the semantic header from a resource automatically. The existing system can deal with four special document formats: HTML, TEXT, LATEX, and RTF.

Since more and more people use PDF for document exchange, perusal on line or in print format due to PDF document's easy to use and cross platform portability, more documents are published in PDF format. This thesis presents the design and implementation of an extension to the existing ASHG to extract the semantic header from a PDF document automatically. First, the PDF document is converted to plain text file using Xpdf, an open source software. Modification to Xpdf has been made to get better results of the conversion. In order to test the accuracy of the ASHG, 500 articles which are all from computer science field are used in an experiment to generate the semantic header; the results of extracting title, abstract, keyword and author name are 92%, 92%, 93% and 80% accurate respectively. However the results reveal that the subject classification (about 41%) is the weakest point of ASHG and requiring further work.

Acknowledgements

I would like to thank my supervisor, Dr. Bipin C. Desai, for his guidance, patience and attentiveness during my study at Concordia University.

I thank the graduate students in the department for their friendship. Also I thank Tao Wang for his help during the experimental stage of the thesis.

Finally, I would like to express my gratitude to my husband Muxin, my lovely son and my parents, for their unconditional love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement

As the amount of information on the web is growing rapidly, it creates new challenge for precise information retrieval. Information retrieval system is to identify and retrieve relevant information from the corpus based on the user query. Due to the large volume of information on the web and poor choice of search terms, too many low quality matches are retrieved as using search engines [1, 2]. In order to produce more relevant search results, a better indexing system of the primary resource is required. Thus, secondary information called meta-information must be extracted and used as an index. Preparing the primary source's meta-information requires finding the primary source, identifying it as to its subject, title, author, keywords, abstract, etc. Also since it is to be used by many users, it has to be accurate, easy to use and properly classified. These problems are addressed by CINDI system (Concordia INdexing and DIscovery System).

1.2 Proposed Solution

The CINDI system, proposed by Dr. Desai et al. [3], is to provide a system for more accurate search and access to the relevant resources on the Internet. In the CINIDI system, the bibliographic database provides information either from contributors' document files or the document files download from the World Wide Web (WWW) site

by a robot which selects trusted sites and uses filters to eliminate non-relevant resources. For cataloguing and searching, CINDI uses a meta-data description called Semantic Header (SH) to extract summarized relevant information from the documents. Since the majority of searches for an information resource begin with a title, name of the authors (70%), subject and sub-subject (50%) [4], therefore the summarized information in SH includes these elements. Also, the abstract and annotations are relevant in deciding whether or not a resource is useful, so they are included in SH too [5, 1]. In order to save resource contributor's time to fill out the semantic header form manually, we built a system called Automatic Semantic Header Generator (ASHG) to generate a draft version of the semantic header for a resource automatically [6]. The existing model for ASHG is available for HTML, Text, Rich Text, and Latex and implemented in the Linux environment.

The Portable Document Format (PDF) is a file format used to represent a document, which is independent of the application software, hardware, and operating system used to create it. Unlike HTML, Text, Rich Text, and Latex, the PDF document consists of non-displayable characters. It uses Adobe imaging model to represent text and graphics. Since a PDF file is cross platform and easy to use, more and more people use it for document exchange and perusal on line or in print format. Consequently, there is a need to support PDF documents in the CINDI system. Hence we propose generating the Semantic Header from the PDF document automatically to incorporate PDF documents in the CINDI system. First, the PDF document is converted to a plain text file by using Xpdf, an open source code software. Modification to Xpdf has been made to get better results of the text

format. The detailed design and implementation is described in chapter 4. In order to test the accuracy of the ASHG, 500 articles, which are all from computer science field, are tested; the results of extracting title, abstract, keyword and author name are presented.

1.3 Structure of the Thesis

The structure of the thesis is as follows. In chapter 2, the backgrounds of the information retrieval, search engine, the CINDI system and the previous work on the ASHG are given. Chapter 3 discusses Xpdf software, which is used to convert a PDF document to a plain text file, and the modifications of Xpdf software are given. Chapter 4 describes in details the design and implementation of extracting title, abstract, keyword, and author name etc. Chapter 5 tests the implemented system and analyzes the experimental results. We draw our conclusions and present some ideas for future works in Chapter 6.

Chapter 2

Backgrounds and Literature Review

2.1 Information Retrieval

Information Retrieval (IR) is a field that studies the problem of finding relevant documents in document collections according to given user queries. Salton and McGill [7] illustrated "Information Retrieval (IR) is concerned with the representation, storage, organization, and accessing of information items". According to the study of Hayes et al. [8], IR research first appeared in the 1960s, in the late 1970s it became a separate discipline of Computer Science. In the 1990s due to the growth of World Wide Web and data storage capacity of computers, the number and size of document repositories increased enormously. As a result, finding useful information on the web is the challenge problem for IR research [2,9].

We use Baeza-Yates et al. [9] software architecture as shown in Figure 2.1 to describe the retrieval process. Before the retrieval process starts, the index structure will be built for the documents. The user submits a query first; then the query is parsed, transformed and processed to retrieve documents. The index structure built previously can make the query processing fast. The retrieved documents are ranked according to the relevance. Finally the user examines the set of ranked documents to get useful information.

Indexes play very important role in modern information retrieval system. Baeza-Yates et al. [9] depicts the role of the index as: "The most important of the tools for information retrieval is the index—a collection of terms with pointers to places where information about documents can be found" and "indexing is building a data structure that will allow quick searching of the text". Among a large variety of methods of IR, keyword-based retrieval is the most-studied and often-used method. The following description of keyword-based IR method is based on the idea of Hayes et al. [8].

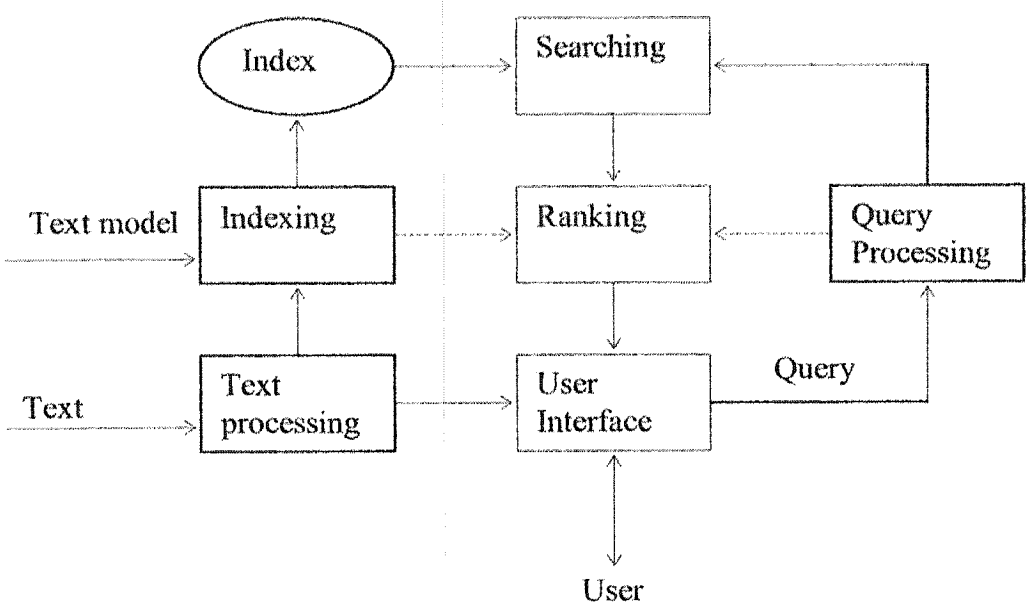

FIGURE 2.1 THE PROCESS OF RETRIEVING INFORMATION

In keyword-based IR, the keywords for each document in the repository are determined first. Also keywords for user queries are analyzed to determine how they match the keywords associated with each document in the collection. In most keyword-based methods, a similarity measure is used to decide the relevance of the document to the

query. The results of the query are given as a list of documents in descending order according to their expected relevance to the query.

The quality of IR methods is measured by how well the retrieved documents match the user's expectations. Precision and recall are two typical metrics to describe the quality of IR methods. Precision is defined as the ratio of the number of relevant documents to the number of retrieved documents:

$$\text{Precision} = \frac{\text{Number of relevant documents}}{\text{Number of retrieved documents}}$$

Recall is defined as the ratio of relevant documents that are retrieved to the total number of relevant documents:

$$\text{Recall} = \frac{\text{Number of relevant documents that are retrieved}}{\text{Total number of relevant documents}}$$

For example, suppose there are 80 documents relevant to widgets in a group of documents that a user wishes to get information from. CINDI system returns 60 documents, 40 of which are about widgets. Then CINDI's precision is 40/60 = 67%, while its recall is 40/80 = 50%. In the chapter 5, when we do the experiments to test our system's accuracy, the feedback on our method's quality is based on the precision metrics.

Between the speed of information retrieval, precision, and recall, there is a three way trade-off model, which is a basic model recognized by web users from traditional information retrieval system. However, this trade-off becomes difficult to balance when the number of documents and web users rapidly increase [2].

2.2 Search Engines

Internet is spreading rapidly since 1995; its scale, coverage, content, and functions provided have developed rapidly. With such an enormous and rich information resource, it is difficult to find something we are interested in without the aid of information navigation tools. About 85% of Web users surveyed claim to be using search engines or some kind of search tool to find the required information [2]. WWW search engine, which is the interactive query tool on the Internet, is gradually becoming the primary tool for information discovery. WWW search engines are expected to handle data collections that are at least 2-3 orders of magnitude greater than a typical test collection for an IR system which is around 5-7 GB, and should provide answers within seconds. Hayes et al. [8] pointed that there are two major differences between classical and web-based retrieval system. First, a classical database is very different from a database for web-based retrieval system. The database of all web pages can be considered as a gigantic database, while in the classical database, its elements can be organized, stored, and indexed to get fast and accurate retrieval. The number of simultaneous users of popular search engines and the number of documents that can be accessed in the web-based retrieval system are the second major difference from the classical retrieval system. The universal problem of the search engine is that too many documents are retrieved and many of them are not

relevant to the user's request. So improving the accuracy of the search results is the primary goal of the search engine [2, 10, 11].

A large study done by Arasu et al. [12] shows that 23% of all the web pages change daily, while 40% of commercial web pages change daily, and some web pages disappear completely. The average half-life time for web pages is 10 days. Therefore, indexing web pages to retrieve information is a complex and challenging problem. A given individual search engine only indexed between 3% to 34% of the possible total web pages estimated by Lawrence and Giles [13]. A follow up study for the top 11 search engines (HotBot, AltaVista, Northern Light, Excite, Infoseek, Lycos, Snap, Microsoft, Google, Yahoo! and Euroseek) indicates that indexing appears to have become more important than ever, since 83% of sites contained commercial content and 6% contained scientific or educational content [13].

There are four approaches to indexing documents on the web: (1) human or manual indexing; (2) automatic indexing; (3) intelligent or agent-based indexing; and (4) metadata, the Resource Description Framework (RDF [14] is used as metadata coding scheme for Web documents), and annotation-based indexing (annotations include more data than metadata to be attached to a Web document [15,16]) [2]. Among the four indexing methods, metadata will be described in details since CINDI system (see section 2.3) is using metadata. Human indexing is done by the experts on popular subjects. The experts organize and compile the directories and indexes to assist the search process. Automatic indexing is carried out with the aid of modern computing equipment.

Intelligent agents are used to perform specific tasks, such as indexing on the Web. They are most commonly referred to as crawlers, ants, automatic indexers, bots, spiders, Web robots, and worms. Metadata is described by Cathro [17] as follows: "An element of metadata describes an information resource or helps provide access to an information resource". On the Internet metadata is attached to a web page that facilitates collection of information by automatic indexers. It has no effect on the visual appearance of the Web page when viewed using a standard Web browser. The Dublin Core Metadata standard and the Warwick framework [18] are two well-publicized metadata standards for Web pages. The Dublin Core is a 15-element metadata set proposed to assist fast and accurate information retrieval on the Internet. The elements are title, creator, subject, description, publisher, contributors, date, resource type, format, resource identifier, source, language, relation, coverage, and rights. The Warwick framework is built on the Dublin results. It provides greater interoperability among resource providers, catalogers and indexers. "The framework is a mechanism for aggregating logically, and perhaps physically, distinct packages of metadata" [18].

The simplest type of metadata for labeling HTML documents is called metatags. Although metadata and annotation could assist fast and accurate search and retrieval, only 34% of homepages use metatags and 0.3% of web sites use the Dublin Core metadata standard [13].

2.3 The CINDI System

There is an urgent need for the development of a system that allows easy search for and access to resources available on the Internet. Solving the problem of fast, efficient discovery and retrieval can be achieved by building a standard index structure and building a bibliographic system using standardized control definitions and terms [5]. A number of systems including WAIS, and a number of Spiders, Worms and other creepy crawlers [19, 20, 21, 22, 23, 24, 25, 26, 27] attempts to provide easy search of relevant documents on the Web. However, due to the large volume of information on the web and poor choice of search terms, their selectivity of documents is often poor [1, 2]. The CINDI system proposed in [3] is designed to solve these problems.

The CINDI system is an information retrieval and indexing system. The objective of the project is to build a system that enables any resource contributor to catalog his/her own resource and any user to search for the resources available from these resources using typical search criteria such as Author, Title, Subject, etc. The system will offer a bibliographic database that provides information about documents available on the Internet by a Web robot. Therefore, secondary information called metadata must be extracted and used as an index to the available primary resource. CINDI uses a metadata description called the Semantic Header to describe an information resource. Desai [28] first introduced Semantic Header in 1994 and Semantic Header predated Dublin Core metadata. Desai [29] pointed out that the current Dublin Metadata Element list suffered

from the absence of the abstract. The later version of Semantic Header was developed with the collaboration of the authors of the Semantic Header [5, 6].

2.4 The Semantic Header

The purpose of semantic header is to include the most often searched elements for an information resource. The statistics done by Katz [4] show that 70% of searches begin with a title and author name, 50% for the subject and sub-subject. Therefore, the elements of the semantic header include the following items:

Title, Alt-title

Title is the required field. It is a name given to the resource by its creator. The alternate title field is an optional element and used as a secondary title of the resource.

Subject (3-level subject hierarchy)

Subject field contains a list of possible subject classifications of the resource. The subject and sub-subject is a repeating group, which is a multi-part field with one or more occurrences of items in the group. At least one entry is required

Author and other responsible agents

For the author and other responsible agents (editor, compiler), the information includes fields such as name, telephone number, fax number, and email address. At least the name or the organization and address is required

Keyword

Keyword is the required field. This field contains a list of keywords used in the resource.

Abstract

The abstract of the documents is either provided by the author or generated by ASHG.

Identifier

Examples of identifiers are ISBN (International Standard Book Number), URL (Universal Resource Locator) of the document. This is the required field. This is a multi-valued slot in case the document is available in many formats or is electronically stored at more than one site.

Date

Date is the required field. The date(s) on which the document was created, catalogued, and the date on which the document will expire, if any.

Version

The version number is given in this element.

Classification

Examples of classification are the legal, security or other type. For each, nature of classification is specified.

Coverage

It indicates the targeted audience of the document or it may indicate cultural and temporal aspect of the document's content.

System Requirements

The electronic document requires certain system requirements to be displayed or used. Examples of the system requirements are the hardware or software platform or the network requirements. For each, the components and the corresponding requirements are specified.

Genre

It is used to describe the artistic, physical or electronic format of the resource. It consists of a domain and the corresponding value.

Source and Reference

The Source indicates the documents being referenced or which were required in its preparation. It could also be the main component for which the current document is an addendum or attachment.

Cost

If there is a fee for the resource, the cost of accessing is given.

Annotations

Annotations put in by readers of the document.

User ID, Password

A Provider ID of at least six characters and a password of four to eight characters. More than one semantic header by the same provider can have the same ID and password.

2.5 Comparision of the Dubin Core Metadata with Semantic Header

The Dublin Core Metadata standard (details see section 2.2) is a well-publicized metadata standard for Web pages; while Semantic Header (details see section 2.4) is used in the CINDI system as a metadata description to describe an information resource. Since Semantic Header requires a minimum set, it cannot be an empty data file. While all the elements in Dublin Core Metadata are optional, hence the whole data file of the Dublin Core Metadata may be empty. Including abstract (an optional but a recommended element) and annotations, Semantic Header provides the more indicative of the contents

for an information resource than the title or keywords. It also includes cost as an optional element since the user may need extra information of the resource. Table 2.1 lists the comparison of the Semantic Header with the Dublin Core Metadata.

Table 2.1 Comparison of the Dublin Core Metadata with Semantic Header

	Dublin Core Metadata	Semantic Header
Year	1995 (the first workshop)	1994 (the first proposal)
Total elements	15	20
A minimum set required	no	yes
Provding abstract	no	yes
Provding annotations	no	yes
Providing cost	no	yes
Verifying by contributor	no	yes

Therefore, the advantages of semantic header are as follows. Semantic header is written in SGML (Standard Generalized Markup Language) format, therefore it allows user to access the indexation of resources online or offline by the Internet web browsers. Many search engines do not provide the abstract of the target resources, semantic header does. Semantic header may become a part of each document. In the CINDI system, the contributor of the resource has to register first and verify the semantic header information of the resource being placed in the system before the resource can be stored into the virtual library. By doing this, it will improve accuracy and efficiency.

2.6 Automatic Semantic Header Generator System (ASHG)

ASHG is used to generate a draft version of the semantic header for a resource automatically and thus save resource contributor's time in filling out the semantic header form manually. The whole procedure of generating SH automatically is described here. In the registering sub-system in CINDI, a prospective contributor who wants to upload his/her document into CINDI system is required to register into CINDI system by filling out a registration form. Once all the required information is provided, CINDI system will register the contributor and a user name and password would be emailed to the prospective contributor. Now the contributor can login to the CINDI system to upload a document and register the semantic header of the new document. First the contributor uploads the document. Next, depending on the file type of the document, the contributor chooses corresponding semantic header extractor (web interface in Figure 2.2). The ASHG system will process the document and generate a draft SH which will be displayed to the contributor via a web page (shown in Figure 2.3). If necessary, the contributor corrects/modifies the SH and click ACCEPT button to confirm it (shown in Figure 2.4). The contributor also can update an existing semantic header later. Since the contributor is required to verify and correct the draft semantic header, therefore its accuracy is much higher than most automatically generated indexes.

The existing system can deal with four document formats: HTML, TEXT, LATEX, and RTF. Since more and more people use PDF for document, hence, it becomes urgent to add PDF document format into the ASHG system. In the next chapter, PDF file is

introduced first, and then the several PDF converters, which extract text file from a PDF file, are discussed. Our solution to PDF conversion is given in the final part of the chapter.

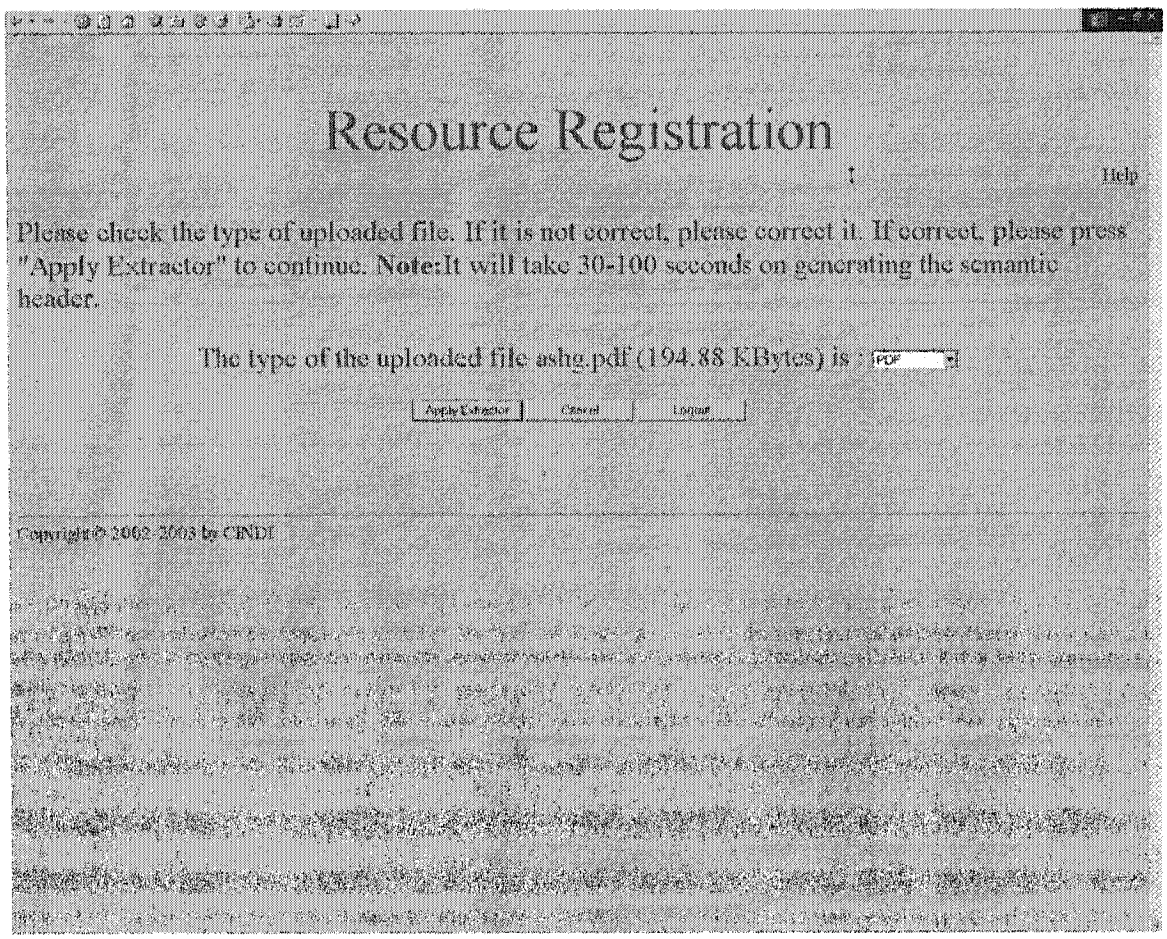

FIGURE 2.2 UPLOAD THE DOCUMENT

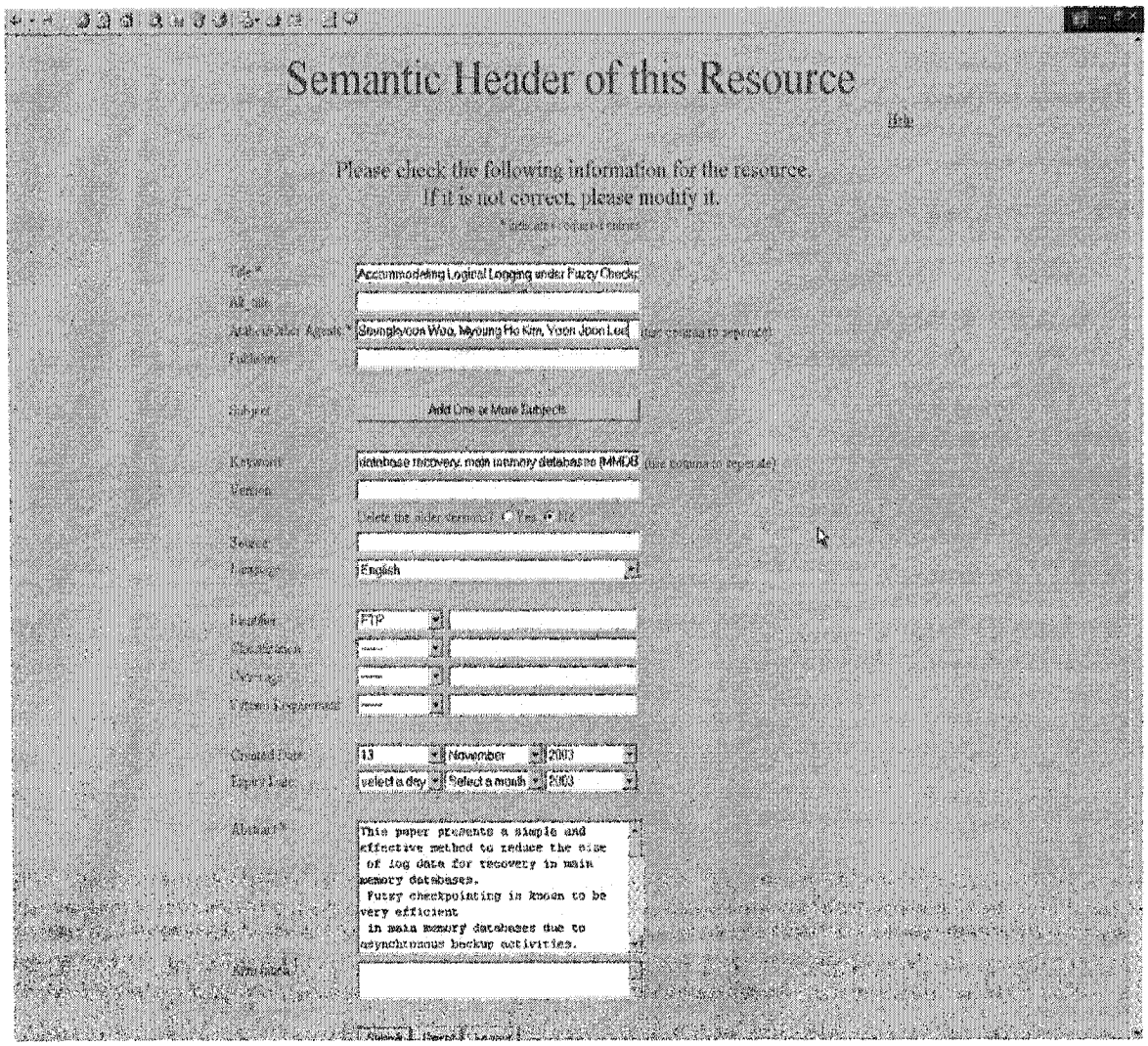

FIGURE 2.3 THE GENERATED SEMANTIC HEADER

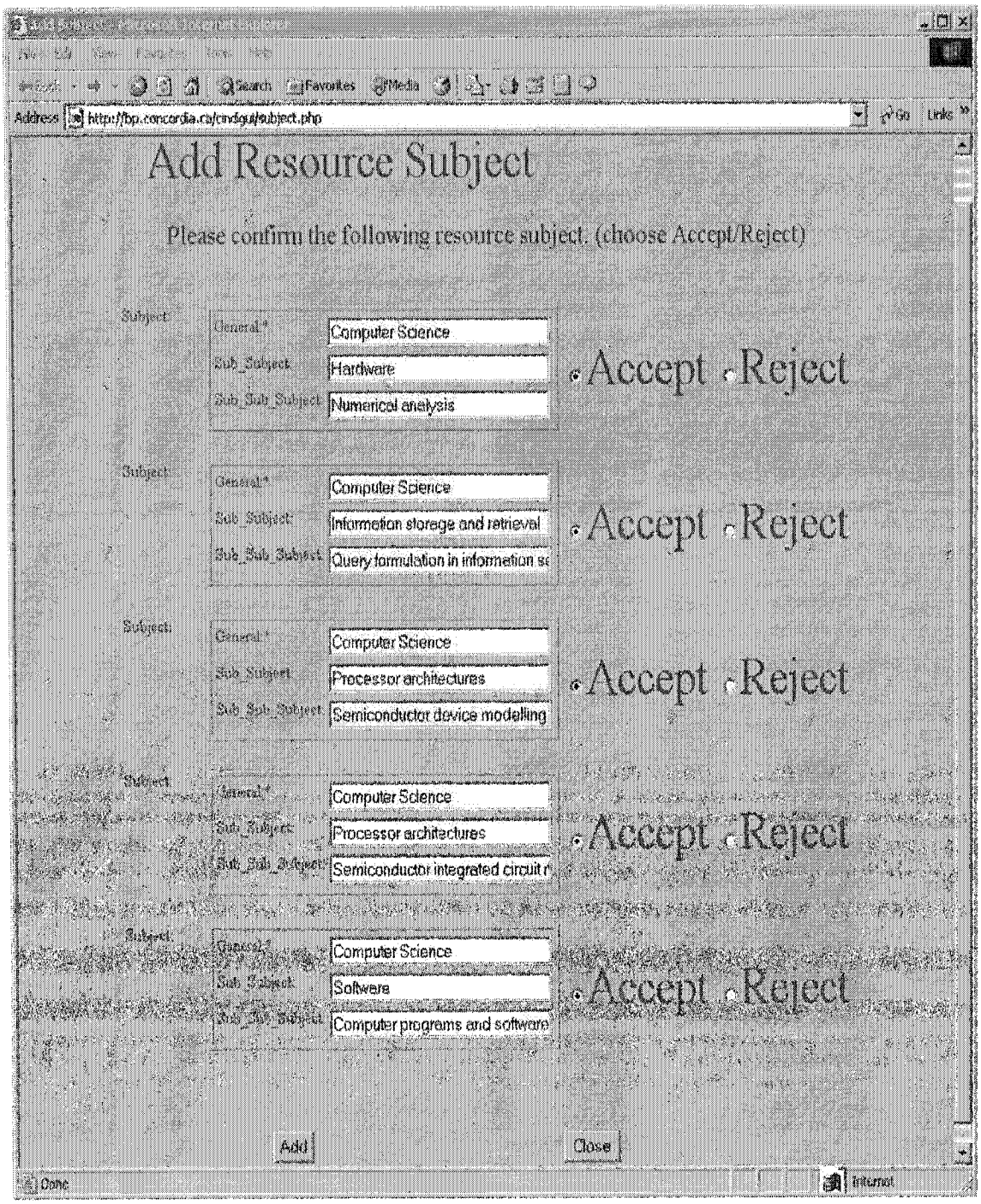

FIGURE 2.4 CONFIRM THE RESOURCE SUBJECT

Chapter 3

Converting a PDF Document to a Text File

The organization of this chapter is as follows: first we introduce the PDF file itself, then several PDF converters which extract text from the PDF file are compared, details are given for Xpdf and pstotext. Finally, a modification for Xpdf is given.

3.1 Introduction of PDF

The Portable Document Format (PDF) is a file format used to represent a document in a manner independent of the application software, hardware, and operating system used to create it. Unlike HTML, PDF files do not contain plain text. It represents text and graphics using the Adobe imaging model. PDF is a page description format and contains one or more pages. Each page in the document may contain any combination of text, graphics, and images.

PDF consists of four parts: objects, file structure, document structure and page description, illustrated in Figure 3.1 [30]. Objects is the set of basic object types used by PDF to represent objects, such as Booleans, numbers, strings, names, arrays, dictionaries, and streams. File structure determines how objects are stored in a PDF file, how they are accessed and how they are updated. It is independent of the semantics of the objects. Document structure specifies how the basic object types are used to represent components of a PDF document, such as pages, annotations, and fonts. A PDF page

description has only limited interaction with other parts of a PDF document. It is part of a PDF page object.

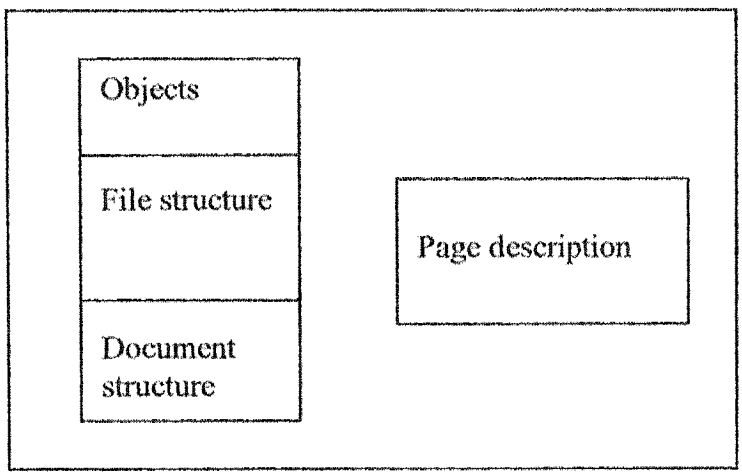

FIGURE 3.1 PDF COMPONENTS

A PDF file has the following advantages:

- **Cross multiple platforms**: PDF file is independent of the software, hardware, and the operating system used to create the file.
- **Easy to navigate**: PDF users and creators can use internal and external links, bookmarks, thumbnails of each page, article threads, etc, to view and find the PDF contents easily.
- **Easy to view:** PDF file has a precise color match regardless of the monitor used, therefore PDF file allows the user to magnify documents up to 800% without the loss of clarity in text or graphics.
- **Smaller file size:** PDF file can be optimized to reduce its file size. For example, PDF file can be 1/5 of the size of its HTML counterpart and 1/4 of the size of its DOC type counterpart.

- **Easy to use:** PDF file can be viewed within Netscape and Internet Explorer windows and saved for offline use or printed.
- **Incremental update:** Developers can create their own software to read, create or modify PDF files without special permission or licensing.

Since the PDF document consists of non-displayable characters, we need to extract the text from the PDF file first, then generate the semantic header from the text file. In this chapter, a number of PDF converters, which extract text from the PDF file, are listed, among them pstotext and Xpdf are introduced in details. Due to the drawbacks of these two products, a modification is made to Xpdf in order to get better-converted text format.

3.2 PDF Converters

Searching the web site, one can find a few software products, which converts PDF documents into ASCII text format. Table 3.1 lists some of these software products. We focus on the two open source software on Linux platform since Linux is the working environment for this project: Xpdf software and pstotext developed by the Digital Equipment Corporation.

Table 3.1 The list of software products

No.	Company	Product	Operating system	Price*
1	Intelligent Converters software	PDF-to-Text	Microsoft Windows	$29.85
2	PDF TO ALL	Pdf-Converter	Microsoft Windows 95/98/2000/NT/ME/XP	$199
3	ConvertZone	CZ-Pdf2Txt	Microsoft Windows 95/98/2000/NT/ME	$199
4	Square One	pdf2text	Windows and UNIX	Unix single license $349
5	Retsina software solutions	PDF Plain Text Extractor v.2.3	Microsoft Windows 98/2000/NT/ME/XP	Source Code $1299.95
6	Traction Software	PDF2Text	Microsoft Windows 95/98/2000/NT/ME/XP	$49.95
7	Digital Equipment Corporation	*pstotext*	Unix, Windows, OS/2, and VMS	free
8	Xpdf software	pdftotext	Windows and Linux	free

Note: Price* is as of date November 2003

3.2.1 Introduction of pstotext software

pstotext is a program developed by the Digital Equipment Corporation. It works with Ghostscript (version 3.33 or later) to extract plain text from PDF files (the system should have Ghostscript 3.51 or later for PDF).

First, pstotext loads a PostScript library from Ghostscript then writes to its standard output information about each string rendered by a PDF document. This information includes the characters of the string, and the information of how to approximate the string's bounding rectangle. Then it post-processes this information and outputs a sequence of words delimited by space, new line, and form feed. pstotext outputs words in the same sequence as they are rendered by the document. Within this sequence, words are separated by either space or new line depending on whether or not they fall on the same line. Each page is terminated with a form feed.

In order to get correct spacing between particular pairs of characters, a PDF document often renders one word as several strings. pstotext uses a simple heuristic to assemble these strings back into words: strings separated by a distance of less than 0.3 times the minimum of the average character widths in the two strings are considered to be part of the same word. However, this typically causes leading and trailing punctuation characters to be included with a word. pstotext translates to the ISO 8859-1 (Latin-1) character code, which is an extension to ASCII covering most of the Western European languages. However, there are mistakes when encoding vector doesn't follow Adobe's conventions.

3.2.2 Xpdf's pdftotext Features

Xpdf is an open source viewer for Portable Document Format (PDF) files. Xpdf's pdftotext program is used to convert a PDF document to a text file. It runs under the X Window System on UNIX, VMS, and OS/2. The tool pdftotext also can run on Win32

systems and any system with a decent C++ compiler. By default, Xpdf will use X server fonts. It requires the following fonts: Courier, Helvetica, Times, and Zapf Dingbats.

Each page in a PDF file is defined by a content stream(s) containing a series of commands. These commands change the current color, draw filled polygons, change the current font, draw text, and so on. For example, the following PDF file displays a single line of text consisting of the string "Hello World" in 24-point Helvetica.

```
%PDF.1.4
1 0 obj
<< /Type /Catalog
/Outlines 2 0 R
/Pages 3 0 R
>>
endobj
2 0 obj
<< /Type /Outlines
/Count 0
>>
endobj
3 0 obj
<< /Type /Pages
/Kids [4 0 R]
/Count 1
>>
endobj
4 0 obj
<< /Type /Page
/Parent 3 0 R
/MediaBox [0 0 612 792]
/Contents 5 0 R
/Resources << /ProcSet 6 0 R
/Font << /F1 7 0 R >>
>>
>>
endobj
5 0 obj
<< /Length 73 >>
stream
BT
```

```
/F1 24 Tf
100 100 Td
(Hello World) Tj
ET
endstream
endobj
6 0 obj
[/PDF /Text]
endobj
7 0 obj
<< /Type /Font
/Subtype /Type1
/Name /F1
/BaseFont /Helvetica
/Encoding /MacRomanEncoding
>>
endobj
xref
0 8
0000000000 65535 f
0000000009 00000 n
0000000074 00000 n
0000000120 00000 n
0000000179 00000 n
0000000364 00000 n
0000000466 00000 n
0000000496 00000 n
trailer
<< /Size 8
/Root 1 0 R
>>
startxref
625
%%EOF
```

Text can be broken into small chunks for purposes of kerning. To display the string "Text can be..." a PDF file might draw "T", and then move back a little to the left, draw "ext can be...". A PDF text extractor must reassemble this into the proper sequence of characters. A PDF file can use any number of fonts. A font is a collection of glyphs - Times-Roman, Helvetica, and Courier. For example, each font has its own glyph for the

letter ‘A’. Also it has an encoding, which is a mapping from character codes (numbers) to glyph names.

There's no rule that requires use of a standard ASCII encoding, nor is there any rule that requires use of standard glyph names (such as 'period', 'zero', and 'A'). If a font contained a glyph named 'Alice' for the letter 'T', a glyph named 'Bob' for the letter 'h', and a glyph named 'Charlie' for the letter 'e', and the font's encoding mapped code 97 to 'Alice', code 14 to 'Bob', and code 53 to 'Charlie', then a string containing the code sequence (97, 14, 53) would generate the word 'The' on the screen or printer.

When PDF generation software creates font subsets to make the PDF file smaller, and also makes it harder to pirate the font, it creates a difficult situation to extract text correctly. A font subset is a font that contains only the glyphs actually used in the document. For example, if a PDF only used 'T', 'h', and 'e' in a particular font, the PDF generator might create a subset font containing just those three glyphs. And it might rename the glyphs 'p01', 'p02', and 'p03', and encode them as codes 1, 2, and 3. In this situation, it is impossible to get the text back out of the PDF file. Therefore, it is hard to extract text from a PDF file with 100% accuracy.

However, normally there is enough information to use font subsets in a PDF file. For example, the glyph names often contain the original character codes: the ASCII code for 'T' is 84, and the font subset might use 'p84' as the glyph name for this character. Additionally, some latest version PDF files provide 'ToUnicode' tables for their fonts.

These map character codes straight to Unicode, avoiding all of the problems with encodings and glyph names.

The result of converting a PDF document to text file is satisfactory when there is only one-column text in the PDF document. However, when the document consists of figures, tables, equations and images, the converted text file will not have the same contents as the PDF document. This is because text can only display the characters from the ISO-8859-1 character set (the advantage of the text file is to be read on any type of computer using any operating system). For figures, the converted text file will lose the lines, curves etc. and only text appearing in the figures would be retrieved. For tables, the converted result will lose the alignment of the columns and rows; therefore the converted table's text may not be easy to understand or convey the same significance. For equations, the converted result will almost totally lose the original meaning, since text file doesn't support special character used in typical equations, such as the symbol for square root, the symbol for integration. For image of .gif or .jpeg format, the converted result is an empty line.

For two-column or more than two-column PDF document, there is a problem in the original downloaded Xpdf code. It concatenates the last line in the first column with the first line in next column without a space. For example, "have" is the last word in the column one, and "created" is the first word in the column two, Xpdf converts these to give: "... havecreated ... "

Since in the typical two-column PDF document, the title, keywords and abstract mostly are in one column format. The concatenation problem may cause no problem in converting this. However, it will have impact on the author information's extraction. For example, when there is more than one author writing the article, the author information are distributed in two or more columns. This may cause errors. For example, the last line in the first column is: "Z.Stojanovic@its.tudelft.nl ", the first line in the second column is "Slobodanka DjordjeviF-Kajan," the converted result would be: Z.Stojanovic@its.tudelft.nlSlobodanka DjordjeviF-Kajan,.

3.2.3 The difference between pstotext and pdftotext

Both pstotext and Xpdf can run on most Unix, Windows, OS/2, and VMS. Also both of them can convert two columns PDF file into text file with the same sequence as rendered by the document. Compared with Xpdf, the converted text file from PDF file by using pstotext has the following drawbacks:

- For the large space between words, the results of pstotext for resembling a word are not as good as Xpdf
- For some European names and Special characters in the PDF document such as **ä, ö, Ü, á, à, ê, ç, ø, å**, pstotext cannot convert them correctly at all. Xpdf is able to do better.
- Pstotext doesn't insert extra new lines to separate paragraphs. Therefore it is hard to distinguish title section, author information section, abstract section, and document body. This causes a problem for ASHG.

Therefore, we choose Xpdf as the converter to extract text from a PDF file.

3.3 The Modifications Made to the xpdf-2.01

We have used many versions of Xpdf since its implementation. The earliest version used was Xpdf-1.0, the most recent version available and used is Xpdf-2.02. Many PDF documents have been tested, however, none of them produced satisfactory results. Comparing Xpdf-2.01 and Xpdf-2.02, the converted result in the text file using version Xpdf-2.01 is more satisfactory than Xpdf-2.02 when considering the title, the superscript / subscript in the author name, the section separation and etc. Therefore, the modifications are based on the xpdf-2.01 instead of Xpdf-2.02. The problems appeared in the converted text files when using Xpdf-2.01 are as following:

1. For large space between each word in the original PDF file, all words stick together without spaces between words in the converted text.
2. For two-column PDF documents, the last line in first column is concatenated with the first line in next column.
3. There is no space between word and superscript / subscript.

By using Linux command diff to compare the content of program TextOutputDev-modi.cc and TextOutputDev-ori.cc (TextOutputDev-modi.cc is the modified code, TextOutputDev-ori.cc is the original program of Xpdf), the results is shown below:

```
*** TextOutputDev-modi.cc      2003-10-09 11:51:26.000000000 -0400
--- TextOutputDev-ori.cc       2003-10-09 11:44:48.000000000 -0400
***************
```

```
*** 1488,1514 ****
    }

    // look for a superscript
!   if ((fontSize1 > lineMinSuperscriptFontSizeRatio * fontSize0 &&
        fontSize1 < lineMaxSuperscriptFontSizeRatio * fontSize0 &&
        (word->yMax < lastWord->yMax ||
         word->yBase < lastWord->yBase) &&
        word->yMax - lastWord->yMin > lineMinSuperscriptOverlap * fontSize0 &&
!       dx < fontSize0 * lineMaxSuperscriptDeltaX) ||
!    // return dx;
!   //}

    // look for a subscript
!   (fontSize1 > lineMinSubscriptFontSizeRatio * fontSize0 &&
        fontSize1 < lineMaxSubscriptFontSizeRatio * fontSize0 &&
        (word->yMin > lastWord->yMin ||
         word->yBase > lastWord->yBase) &&
        line->yMax - word->yMin > lineMinSubscriptOverlap * fontSize0 &&
!       dx < fontSize0 * lineMaxSubscriptDeltaX)) {
!
!       double space = dx; //added by xue v.2.02
!       return fabs(word->yBase - line->yBase); //added by xue v.2.02
!
!    // return dx;
    }

    return -1;
--- 1488,1510 ----
    }

    // look for a superscript
!   if (fontSize1 > lineMinSuperscriptFontSizeRatio * fontSize0 &&
        fontSize1 < lineMaxSuperscriptFontSizeRatio * fontSize0 &&
        (word->yMax < lastWord->yMax ||
         word->yBase < lastWord->yBase) &&
        (word->yMax < lastWord->yMax ||
         word->yBase < lastWord->yBase) &&
        word->yMax - lastWord->yMin > lineMinSuperscriptOverlap * fontSize0 &&
!       dx < fontSize0 * lineMaxSuperscriptDeltaX) {
!     return dx;
!   }

    // look for a subscript
!   if (fontSize1 > lineMinSubscriptFontSizeRatio * fontSize0 &&
        fontSize1 < lineMaxSubscriptFontSizeRatio * fontSize0 &&
```

```
      (word->yMin > lastWord->yMin ||
       word->yBase > lastWord->yBase) &&
      line->yMax - word->yMin > lineMinSubscriptOverlap * fontSize0 &&
!     dx < fontSize0 * lineMaxSubscriptDeltaX) {
!   return dx;
  }

  return -1;
***************
*** 1865,1875 ****
       (*outputFunc)(outputStream, space, spaceLen);
      }
    }
- //the following added by xue
- //add a space when large space appears in the pdf file
-    if (rawOrder && col < line->col[0]) {
-     (*outputFunc)(outputStream, space, spaceLen);
-    }

      // print the line
      for (i = 0; i < line->len; ++i) {
--- 1861,1866 ----
***************
*** 1879,1896 ****
      col += line->convertedLen;

      // print one or more returns if necessary
! //FOLLOWING COMENT OUT BY xue since I don't want new line between columns
! /*
!     if ((rawOrder && line->len >20) ||
!       !line->pageNext ||
        line->pageNext->col[0] < col ||
        line->pageNext->yMin >
          line->yMax - lineOverlapSlack * line->fontSize) {
- */
-     if (!line->pageNext ||
-       line->pageNext->col[0] < col ||
-       line->pageNext->yMin >
-         line->yMax - lineOverlapSlack * line->fontSize) {

      // compute number of returns
      d = 1;
--- 1870,1879 ----
      col += line->convertedLen;

      // print one or more returns if necessary
```

```
!     if (!line->pageNext ||
        line->pageNext->col[0] < col ||
        line->pageNext->yMin >
          line->yMax - lineOverlapSlack * line->fontSize) {

      // compute number of returns
      d = 1;
```

After the modification, the results in the converted text file are: for large space between each word in a PDF file, now there is a proper space between words for most of the cases (in the original codes all words stick together without spaces). For two-column PDF documents, there is new line between the last line in the first column and the first line in next column (the two line concatenated in the original codes). Also there is a space between word and the superscript / subscript. For example, when author name is "Chantal Reynaud[a] ", the modified codes converts it as "Chantal Reynaud a". While the original codes convert it as "Chantal Reynaud*a* ", it is hard to distinguish the name "Reynaud" and the letter "a" when searching for the author name.

During the program-coding period, it was found that if we inserted a new line between the last line in the first column and the first line in next column, there was a bad effect on title and author name. Since for a two-column PDF file, if the title and author information are in one column format, there is a possibility to break the title line in half and author name in half. This would cause problems when retrieving the title and the author name. Therefore, a space is added between the last line in the first column and the first line in next column instead of inserting a new line (The codes for adding a new line is commented out in the diff results).

Chapter 4

Implementation

A PDF file uses Adobe imaging model to represent text and graphics, therefore a PDF file is made up of non-displayable characters. In order to extract Semantic Header from a PDF document, the PDF file is converted to a text file first. Extracting information from unstructured text file is a challenging task [31, 32]. However, PDF_extractor will try to extract the title, the abstract, the author name, the keywords, the subject and sub-subject etc.

4.1 Illustrate Title, Abstract and Keyword Extraction

ASHG needs to extract the title, keywords, abstract. Title is the required field. It is a name given to the resource by its creator. Abstract is the optional field but it is recommended. Since abstract covers the main points of the article, readers use abstract to see if an article interests them or relates to the topic they're working on. Rather than tracking down hundreds of articles, readers rely on abstract to decide quickly if an article is pertinent. Equally important, readers use abstract to help them gauge the sophistication or complexity of a piece of writing. If the abstract is too technical or too simplistic, readers know that the article will also be too technical or too simplistic. With so many indexes now available electronically, abstract with their keywords are even more

important because readers can review hundreds of abstract quickly to find the ones most useful for their purpose. Keywords might be either explicitly stated or they have to be implicitly extracted.

The syntax used in ASHG to govern the extraction is shown below:

1. < Title > :- Explicitly-Stated-Title
2. < Abstract > :- Explicitly-Stated-Abstract
3. < Abstract > :- < Implicitly-Stated-Abstract >
4. < Implicitly-Stated-Abstract > :- The-First-Paragraph-of-Introduction
5. < Keywords > :- Explicitly-Stated-Keywords
6. < Keywords > :- < Implicitly-Stated-Keywords >
7. < Implicitly-Stated-Keywords > :- <Title> or <Abstract> or <Other-words>

For the < Implicitly-Stated-Abstract >, here is the difference from the previous work of ASHG [6]. In the previous work of ASHG, the syntax for extracting < Implicitly-Stated-Abstract > is as following:

< Implicitly-Stated-Abstract > :- < Keywords > or < Title > or < Other-words >

We have tested hundreds of documents by using previous syntax to extract implicitly stated abstract. The resulting abstract consisted of some words, phases, and sentences together with the syntax stated and is hard to understand. Therefore we use the first paragraph in the introduction to replace the old syntax. The first paragraph of introduction normally is the summary of the whole article, therefore using it as the alternate abstract is better in case the document doesn't have an abstract.

4.1.1 Title Extraction

Normally, the first sentence in the converted text is considered as the title. However, this is not so in all cases especially when a PDF document is converted to text file. The variations are shown below based on our experiments of converting many PDF document:

1. The first line of the text file is the page number
2. The title consists of two lines or more
3. The first few lines is the journal heading, such as "VLDB Journal,4, 567-602 (1995), Stanley Y.W. Su, Editor 567"
4. The title consists of more than one line, however there are empty new lines in between.
5. There is no empty new line between title and author information.

Therefore, we first look for the variations of title and extract the Title according to the most likely variation. If none of the cases apply, then we look for the first continued lines from the text file and it will be assumed to be the title.

4.1.2 Abstract Extraction

Most articles have an explicit abstract. The abstract part can be one paragraph or more. Within an article, the position of abstract usually follows the information for the

author(s). The explicit marked word "Abstract" or "Summary" may be used. The ending position of abstract is usually prior to the keyword or the introduction heading.

In ACM journals, abstract is not explicitly marked with the word "Abstract" or "Summary" for the actual abstract contents in some articles. However, there are still indicators which can be used to decide if there exists an abstract. Following the abstract the article would have the "tag" words such as "Categories and Subject Descriptors ".

Therefore, we look for the explicit abstract first, and if found we select everything until we met the keyword or the introduction heading, or we select the paragraph before the marked works "Categories and Subject Descriptors: ". Otherwise, we will be obliged to select the first paragraph of the introduction.

4.1.3 Keyword Extraction

The explicitly stated keyword has "tag" word(s) such as "Keywords", "Keyword", "Key words", "Additional Key Words and Phrases" and "Index Terms". Therefore, we look for these patterns first; if found we extract the text following the pattern until an Introduction heading or a new paragraph is reached. If the pattern is not found, we assume there are no explicitly stated keywords in the article; in this case, ASHG generates a list of most significant words. First, the system look for words in the title, abstract and other tagged words, then it filters out the noise English words. The remaining words will pass the stem process to remove all suffixes. All the words can be found in keyword database will be

chosen, and the rest are dropped. For every chosen word we will assign weights to each. The word which appears in the abstract, title, and other tagged words are assigned weight of 4, 3, 2 respectively. We extract the words having the highest weight as the keyword. The detail information is in Haddad's thesis section 5.3.5[6].

4.1.4 Phone Number/Fax and Email Extraction

Not every article has fax and/or email information for the author(s), but we still try to extract these information. The pattern for the email is explicitly marked by the character "@", and the phone number/fax is made up of a series of digital numbers. The position of phone number/fax and email would be before the abstract part or introduction part if the abstract is not there.

4.2 Variations of Author Name

Identifying author names is still an open problem [31]. First we will examine the variety of author name format appearing in different text files converted from PDF documents. By collecting these author names from different documents, we got the following information:

1. Name separated by spaces

   Anca Vaduva Klaus R.Dittrich

2. Name separated by ","

   C. Amanatidis, M. Halkidi, M. Vazirgiannis

3. Name separated by "," with a superscript number or letter (such as Chantal Reynaud[a]) attached

   1 Dragan Stojanovic, 1 Slobodanka Djordjevic-Kajan, 2 Zoran Stojanovic

   Note: the original line in PDF document is as shown below:

   [1]Dragan Stojanovic, [1]Slobodanka Djordjevic-Kajan, [2]Zoran Stojanovic

4. Name separated by spaces and ","

   Holger Schwarz , Ralf Wagner , Bernhard Mitschang

5. Name separated by space with a superscript of "* "attached

   Kai-Uwe Sattler* Eike Schallehn**

6. Name separated by spaces and "*"

   Kai-Uwe Sattler * Eike Schallehn

7. Name separated by "and"

   Mingchun Liu and Chunru Wan

8. Name separated by " - "

   Mirella Moura Moro – Silvia Maria Saggiorato – Nina Edelweiss – Clesio Saraiva dos Santos

9. Name separated by "," and "and"

   Niculae Stratica, Leila Kosseim and Bipin C. Desai

10. Name separated by "&"

    P. Chountas & I. Petrounias

11. Name separated by new line

    Spiros Sirmakessis

    Athanasios Tsakalidis

From the above information, it is hard to use a single pattern to extract author name from the PDF documents. Searching for name is still in research process, the task here is to search for author name within an article itself.

4.3 Name Database Built From DBLP

In order to improve the retrieval of author name, we consider using a string match with a database of published author names. The Digital Bibliography & Library Project (DBLP) developed by Dr. Michael Ley, is a free database and currently lists more than 450,000 articles.

The DBLP server provides bibliographic information on major computer science journals and proceedings. Initially the server was focused on Database systems and Logic Programming, now it is gradually being expanded toward other fields of computer science. Only the list of authors, the title, and the publication context (journal, volume, pages, etc) are mandatory. Many papers indexed by DBLP have been published only in printed journals or proceedings. The journals include: CACM, TODS, TOIS, TOPLAS, DKE, VLDB J., Inf. Systems, TPLP, TCS. The conferences include: SIGMOD, VLDB, PODS, ER, EDBT, ICDE, POPL, and IDEAS.

The bibliographic records of DBLP has the following structure:

<**inproceedings** key="BertinoCS98">
<**author**>Elisa Bertino</**author**>

<author>Barbara Catania</author>
<author>Boris Shidlovsky</author>
<title>Towards Optimal Indexing for Segment Databases.</title>
<pages>39-53</pages>
<year>1998</year>
<booktitle>EDBT</booktitle>
<url>db/conf/edbt/edbt98.html#BertinoCS98</url>
</inproceedings>

Since DBLP bibliographic records fit into the XML framework. We can use a C program called dblp_convert.c to extract author names from the DBLP, then insert them into a table called "nameDB" in ASHG database. However, sometimes the author tag contains a HTML escape code like ü, á and etc. The HTML escape code has to be converted to corresponding special character first, then inserted into nameDB database since there is no HTML escape code in the converted text file and the special characters can be shown in the converted text file. For example string Müller in DBLP bibliographic records matches with Müller in nameDB; Stefan Böttcher should be converted to Stefan Böttche then inserted into nameDB database. The relationship between the special character and corresponding HTML escape code is given in Appendix A. The example for a HTML escape code in DBLP bibliographic records is shown below:

<article mdate="2003-01-31" key="tr/ibm/IWBS71">
<author>Stefan Böttcher</author>
<author>Christoph Beierle</author>
<title>Database Support for the PROTOS-L System</title>
<journal>IWBS Report</journal>
<volume>71</volume>

<year>1989</year>

<publisher>IBM Germany Science Center, Institute for Knowledge Based Systems

</publisher>

</article>

Since most of the articles used by ASHG are in computer science field, the database nameDB will cover most of the author names found in the PDF documents currently being targeted. The description of the table nameDB is as followings:

Table 4.1 The description of the table nameDB

Column Name	**Description**
last_name	Primary key
first_name	Primary key

The syntax of creating table nameDB is shown below:

```
Create table nameDB(
        last_name       varchar(40)     not null,
        first_name      varchar(40)     not null,
        primary key(last_name, first_name(10))
) type=innodb;
```

To make the retrieval of a tuple given its primary key value efficient, we need to create an index with the primary key fields as the search key. Since different authors can have the same last name and also different authors can have the same first name. In order to distinguish authors uniquely and search on the last name efficient, we declare {last_name, first_name(10)} as the primary key. It states that two authors may have the

same last name or first name, but not both; two authors having the same names could not be distinguished.

The data type of column first_name is VARCHAR; indexes can be created that use only part of a column. Because most names usually differ in the first 10 characters, this index should not be much slower than an index created from the entire first_name column only. Also, using partial columns for indexes can make the index file much smaller, which could save a lot of disk space and speed up INSERT operations!

Table nameDB uses InnoDB type since InnoDB provides MySQL with a transaction-safe storage engine. It has commit, rollback, and crash recovery capabilities. InnoDB does locking on row level and also provides an Oracle-style consistent non-locking read in *SELECT*s. These features increase multiuser concurrency and performance.

The Pseudo code for inserting author name into table nameDB from DBLP is as follows:

```
void insert_author(char *filename);
void load_character();
char *replace_character(char name[200]);

int main(int argc, char* argv[])
{
  initialize MySQL;
  if(connect with MySQL)
  {
      delete table nameDB;
      if(delete true)
        print message;
      else
        print error message from MySQL;
      call insert_author(argv[1]);
```

```
        close MySQL;
    }
    else
    {
        print error message from MySQL;
        return EXIT_FAILURE;
    }
    return EXIT_SUCCESS;
}

void insert_author(char *filename)
{
    FILE* file = fopen(filename, "r");
    while((c=fgetc(file))!=EOF)
    {
        if(c!='\n')
        {
            read character by character within the line;
        }
        else
        {
            if (strncmp("<author>", line, 8)== 0)
            {
                copy author name within the author tag;
                remember the last space_position within the name;
            }

            if(there is HTML escape characters in first name){
                strcpy(first_name1, replace_character(first_name1));
            }
            if(there is HTML escape characters in last name){
                strcpy(last_name1, replace_character(last_name));
            }
            copy from the first character of the author name to the position of last
space_position into first name;
            if(space_position == 0)
                copy the name tag into last name;
            else
                copy rest of the name string after the position of last space_position;
            concatenate the first name with the last name;
            insert into table nameDB;
            if(there is error)
                print the error message;
        }//else
    }//while
    fclose(file);
```

```
  return;

}

void load_character()
{
  FILE* file = fopen("latin.txt", "r");
  int i=0;

  while(fscanf(file, "%s %s", spec_code[i].spec_char,spec_code[i].html_char)!=
EOF)
  {
    //printf("%s %s\n", spec_code[i].html_char, spec_code[i].spec_char);
    i++;
  }

  fclose(file);

  return;

}

char  *replace_character(char name[200])
{

   if(there is "&" inside the name){
        remember the position pos1;
        flag =1;
   }
   if (flag ==1){
        copy the HTML escape characters from the name;
        using binary search method to find the corresponding special character and
replace the HTML escape characters;
   }
   else{
        return name;
   }
}
```

In the DBLP bibliographic records, author nametag does not explicitly state the last name and the first name since it only provides the full name. The name parts, such as last name, first name, middle name etc. are separated by space. Therefore, in the dblp_convert.c

program, we assume the last part of the full name is the last name, the remaining part of the name is considered as the first name. For most cases that is the order in the nametag of DBLP bibliographic records. However, for some names, especially the names not from English speaking country, such as Chinese names, this may not be the case. Also some name only consists of the last name since there is no first name part in the DBLP bibliographic records. The reason of separating last name from first name is considered as following. First, there are around 250,000 records in nameDB database, and creating an index would enable searching the database more efficiently. Secondly, when comparing the string with the records in the database it is easy to start from the last name. When last name is the same, then continue the process. Third, if the order of the author name in the text file is not the same as the author name in nameDB, we need to separate the last name from first name, reverse the order then compare them. Fourth, some authors may use variation of their name for different papers, such as Dr. Bipin C. Desai may write his name as B. C. Desai. By comparing the last name and the initials of the first name we can identify the author name from the text file.

4.4 Author Name Extraction

There is some difficulty in retrieval of author name from the text:

1. As shown in section 4.1, there is no fixed pattern for author name.
2. Some name can be both first name and last name, such as Jorge Berra. In database nameDB, Jorge can be founded in both last_name and first_name column. Therefore some time it is hard to tell which part is the last name and which part is the first name.

3. Some publications require the author's last name first followed by the first name. Therefore, in the nameDB database, we will put actual last name in the first name field and the actual first name in the last name field. For example, there is one record about the Chinese name written as "Zhang Lixin" in text file where Zhang is the last name and Lixin is the first name. However, in the nameDB database, last_name field is "Lixin", while the first_name field is "Zhang".
4. Same author may use name's variation in different papers. For example Bipin C. Desai may write his name as "B. C. Desai". Also for the author name "Maria Grazia Fugini" may write her name as " M. G. Fugini" or "Maria G. Fugini" or even as "M.G. Fugini".
5. First name and last name is separated by space, also different author names are separated by space. Therefore it is hard to distinguish each author name even by reading. For example, Anca Vaduva Klaus R. Dittrich.

For the case 2, since author name in both text file and database could have the situation where the last name is followed by the first name, there could be four sub-cases for an author name such as "Zhang Lixin", shown in the Table 4.2:

Table 4.2 Cases for comparing name in the database

Cases	Name in the text file		Name in the database	
	First name	Last name	First name	Last name
1	Zhang	Lixin	Lixin	Zhang
2	Lixin	Zhang	Lixin	Zhang
3	Zhang	Lixin	Zhang	Lixn
4	Lixin	Zhang	Zhang	Lixn

For the case 2 and 3, regardless of the name order is reversed or not, the name in the text file is the same as the name in the database. For the case 1 and 4, we should allow for them in the implementation.

The Pseudo code for the author_ext.c is as following:

```
void readName(char *filename );
int reverseName(char *full_name );
int initialName(char *full_name );
int main(int argc, char* argv[])
{
        mysql_init(&my_connection);          // Mysql initialization
        connect with MySQL;
        call readName(argv[1]);
        mysql_close(&my_connection);         // MySQL close
        return 0;
}

void readName(char *filename )
{
   set variables and initialization;
   FILE *file = fopen(filename, "r");
   char my_filename[ ] = "AUTHOR";
   my_stream = fopen (my_filename, "w");
   while(!feof(file))
   {
        read each line from the file;
        if(line is not empty)
        {
           for(break up the string as at each space)
           {
              if(string = = "ABSTRACT" or "Abstract" or "Introduction")
                end of processing;
              if (string = = "and", "-", ",", "+", "*", "digital number", "single letter")
                continue;
              while( lname[j] is not NULL)
              {
                eliminate "," and "*" in the name string and set flag = 1;
                copy each character lname[j] to last_name[j];
                j++;
              }
              if(first string)
```

```
                copy last_name to firstName;
              else if(not the first string in the name)
                concatenate it with firstName;
              else if(  the following name appears)
                first time just copy, then concatenate with firstName;
              else
                concatenate it with firstName;
              query from MySQL "select first_name, last_name from nameDB where
last_name = firstName";
              while(get db_name row by row from nameDB)
              {
                if(string firstName == db_name)
                    write into file AUTHOR;
                else if( there is a reverse order of the name, call reverseName(full_name ))
                    write into file AUTHOR;
                else if( there is a initial name, call initialName(full_name ) )
                    write into file AUTHOR;
                else
                    continue;
              }//while
            }//for
          }//if
    }//while
    fclose(my_stream);
    fclose(file);
}

int reverseName(char *full_name )
{
    copy the first part of full_name to name as the last name;
    query from nameDB "select first_name, last_name from nameDB where last_name =
name";
    while(get db_name row by row from nameDB)
    {
        reverse the first_name and last_name in the nameDB
        if(reversed db_name == full_name)
            return true;
        else
            return false;
    }
}

int initialName(char *full_name )
{
    copy the last part of full_name to name as the last name;
```

```
   query from nameDB "select first_name, last_name from nameDB where last_name =
name";
   while(get db_name row by row from nameDB)
   {
        if(name_part = 1){
           retrieve the first_name field from nameDB and make it as initial name called
part_name and copy it to ini_name;
           retrieve the first part of first_name field from nameDB and copy it to
part_name1;
           strcpy(ini_name2, part_name1);
           strcat(ini_name2, ".");
        }
        if (name_part>1){
           ini_name = "B. C.";
           ini_name1 = "Bipin C.";
           ini_name2 = "B.C.";
        }
        strcpy(db_name, ini_name);
        strcat(db_name, " ");
        strcat(db_name, l_name);

        strcpy(db_name1, ini_name1);
        strcat(db_name, " ");
        strcat(db_name, l_name);

        strcpy(db_name2, ini_name2);
        strcat(db_name, " ");
        strcat(db_name, l_name);

        if(!strcasecmp(db_name, full_name)|| !strcasecmp(db_name1, full_name) ||
!strcasecmp(db_name2, full_name)){
           flag =1;
           break;
        }
        else{
           flag = 0;
           continue;
        }
   }//while
        if(flag==0)
           return 1;
        if(flag==1)
           return 0;

}
```

However, for the following cases, the program cannot retrieve the name automatically.

- Spelling error either in the nameDB or in the text file
- Error in PDF document converting to text file; for example, Dragan Stojanovic in PDF document will be converted as Dragan StojanoviF in text file. The reason is because a PDF file uses Adobe imaging model to represent text (details see chapter 3).
- Database nameDB does not have the record of the author name shown in the text file. However the DBLP bibliographic records will be updated from time to time and it increases quickly (in April 2003, DBLP listed more than 380000 articles, in June 2003, it has more than 390000 articles). Consequently, the name collection in the nameDB will be more complete.

One of advantages of ASHG is that the SH generated is a draft one and the author would check it if necessary make corrections before being stored in the CINDI database.

4.5 Cooperating with other projects in the CINDI system

In order to allow CINDI system work more completely, we have a web robot, which downloads millions of article from Computer Science field into our system and saved in our database. The PDF converter will filter out the non-scientific articles and convert the article to PDF document if the article is not HTML, Latex, RTX, and Text format. Also there will be an index built for these articles. The schema of the table DOWNLOADS_STATUS describes the information of the downloaded articles, shown below:

```
CREATE TABLE DOWNLOAD_STATUS (
    ID              int(50)          NOT NULL     auto_increment,
    prefix_url      varchar(100)     NULL,
    file_name       varchar(100)     NULL,
    temp_location   varchar(100)     NULL,
    final_location  varchar(100)     NULL,
    ddate           date             NULL,
    size            int(10)          NULL,
    file_type       varchar(20)      NULL,
    pdf_flag        smallint(1)      NULL,
    ashg_flag       smallint(1)      NULL,
    filter_flag     smallint(1)      NULL,

    PRIMARY KEY (ID)

) TYPE = InnoDB;
```

ASHG checks the table DOWNLOADS_STATUS regularly, if there is an article, whose type is pdf, html, latex, rtf, text and ashg_flag = 0, the corresponding semantic header should be generated. Then set the ashg_flag = 1. If the article's type is not the type mentioned above, ashg_flag = 0 and pdf_flag = 1, which means the original article type is not pdf now it is converted to a pdf file, and its semantic header is not generated. ASHG should check the table CONVERT_PDF in another subsystem of CINDI under development to get the file location and file name then generates its semantic header. The schema of table ASHG and PDF_CONVERT are shown below:

```
CREATE TABLE ASHG (
    ASHG_ID         int(50)          NOT NULL     auto_increment,
    docID           int(50)          NOT NULL,
    create_time     datetime         NULL,
    update_time     datetime         NULL,
    ashg_location   varchar(100)     NULL,
    txt_location    varchar(100)     NULL,
    ashg_filename   varchar(100)     NULL,
    txt_filename    varchar(100)     NULL,
    index_flag      smallint(1)      NULL,

    PRIMARY KEY (ASHG_ID),
```

```
    INDEX (docID),
    FOREIGN KEY (docID) REFERENCES DOWNLOAD_STATUS(ID)
    ON UPDATE CASCADE ON DELETE RESTRICT

) TYPE = InnoDB;


CREATE TABLE PDF_CONVERT (
    ID              int(50)         NOT NULL      auto_increment,
    docID           int(50)         NOT NULL,
    filename        varchar(100)    NULL,
    convert_date    date            NULL,
    location        varchar(100)    NULL,

    PRIMARY KEY (ID),
    INDEX (docID),
    FOREIGN KEY (docID) REFERENCES DOWNLOAD_STATUS (ID)

) TYPE = InnoDB;
```

The pseudo codes to generate semantic header from the downloaded articles is as follows:

```
int main () {
  if( connecting with mysql){
  executing query1("SELECT file_type FROM DOWNLOAD_STATUS WHERE
ashg_flag = 0;");
  if(query1 successful){
       while(there is row in the result){
        if(file_type == "pdf"){
          fetch the result;
          run the PDF_extractor for the fetched file;
          if(converted text file and generated ASHG are available){
                insert all the data into ASHG;
               update ashg_flag = 1;
          }
          else{
               print error message;
               delete the text file and the generated ASHG file;
          }
          else if(file_type == "html"){
          print "this is html file.\n";
          fetch the result;
          run the HTML_extractor for the fetched file
```

```
            insert all the data into ASHG;
            update ashg_flag = 1;
          }
          else if(file_type == "latex"){
            print "this is latex file.\n";
            fetch the result;
            run the LATEX_extractor for the fetched file
            insert all the data into ASHG;
            update ashg_flag = 1;
          }
          else if(file_type == "rtf"){
            print "this is rtf file.\n";
            fetch the result;
            run the RTF_extractor for the fetched file
            insert all the data into ASHG;
            update ashg_flag = 1;
          }
          else if(file_type == "txt"){
            print "this is text file.\n";
            fetch the result;
            run the TEXT_extractor for the fetched file
            insert all the data into ASHG;
            update ashg_flag = 1;
          }
          else{
            fetch the result;
            if(converted to PDF already){
                  print "this is pdf file.\n";
                  run the PDF_extractor for the fetched file
                  if(converted text file and generated ASHG are available){
                        insert all the data into ASHG;
                        update ashg_flag = 1;
                  }
                  else{
                        print("something wrong with the converted text file and generated
ASHG.\n");
                        delete the text file and the generated ASHG file;
                  }
            }
          }//else
        }//while
    }//if
    else{
        print error message;
    }
  }//if
```

```
  else{
    print error message;
  }
  close mysql;
}
```

Chapter 5

Tests and Results

In this chapter, we illustrate how the ASHG system extracts the meta-information from PDF documents; also we demonstrate the automatic subject classification of the ASHG. For each PDF document, we apply ASHG and show the results. We compare the results with the original resource and for the subject classifications generated by ASHG we compare them with that of Inspec (The Database for Physics, Electronics and Computing) [33] for the same document.

5.1 Experiments

The experiments were conducted on 500 PDF documents to test the accuracy of the generated index and the subject classification results. There are five generated fields of ASHG to be generated: title, author name, abstract, keywords and subject classification. The 500 PDF articles come from varieties of sources, such as Conferences: SIGMOD, VLDB, PODS, ER. Journals: DATA BASE, TODS. After applying the ASHG on a set of documents, the generated index fields such as title, author name, abstract, keywords are compared with those that are found in the original articles. The ASHG's automatic subject classification results are compared with the Inspec's classification. Table 5.1 will shows the title and the source of the first 20 articles, the complete titles and sources of all

articles are given in the Appendix B. Table 5.2 shows the test results of the first 20 articles, the complete test results for ALL articles are shown in Appendix C.

Table 5.1 The title and the source for the first 20 test articles

PDF No.	Journal /Proceedings	Volume / Year	Testing Paper Title
D001	TODS	Vol 24, 1999	Broadcast Protocols to Support Efficient Retrieval from Databases by Mobile Users
D002	TODS	Vol 24, 1999	Database Design for Incomplete Relations
D003	TODS	Vol 24, 1999	Temporal FDs on Complex Objects
D004	TODS	Vol 24, 1999	Optimization of Queries with User-Defined Predicates
D005	TODS	Vol 24, 1999	GlOSS: Text-Source Discovery over the internet
D006	TODS	Vol 24, 1999	Distance Browsing in Spatial Databases
D007	TODS	Vol 24, 1999	Supporting Valid-Time Indeterminacy
D008	TODS	Vol 24, 1999	Safe Query Languages for Constraint Databases
D009	TODS	Vol 24, 1999	Safe Stratified Datalog With Integer Order Does Not Have Syntax
D010	TODS	Vol 24, 1999	Optimization Techniques for Queries with Expensive Methods
D011	TODS	Vol 23, 1998	Multiview Access Protocols for Large-Scale Replication
D012	TODS	Vol 23, 1998	Ensuring Consistency in Multidatabases by Preserving Two-Level Serializability
D013	TODS	Vol 23, 1998	An Access Control Model Supporting Periodicity Constraints and Temporal Reasoning
D014	TODS	Vol 23, 1998	Conceptual Schema Analysis: Techniques and Applications+N31
D015	TODS	Vol 23, 1998	An Efficient Method for Checking Object-Oriented Database Schema Correctness
D016	TODS	Vol 23, 1998	Information Gathering in the World Wide Web: The W3QL Query Language and the W3QS System
D017	TODS	Vol 23, 1998	Towards a Theory of Cost Management for Digital Libraries and Electronic Commerce
D018	TODS	Vol 23, 1998	Inverted Files Versus Signature Files for Text Indexing
D019	TODS	Vol 22, 1997	Extended Ephemeral Logging: Log Storage Management for Applications with Long-Lived Transactions
D020	TODS	Vol 22, 1997	Outerjoin Simplification and Reordering for Query Optimization

Table 5.2 The test results for the first 20 test articles

PDF No.	Title	Author	Abstract	Keyword	Subject	Converting Problems
D001	1.00	3/4	1.00	1	1/3	0
D002	1.00	1/2	0.50	1	1/4	0
D003	1.00	1/1	1.00	1	2/3	0
D004	1.00	1/2	1.00	1	1/2	0
D005	1.00	1/3	1.00	1	1/4	0
D006	1.00	1/2	1.00	1	1/3	0
D007	1.00	1/2	1.00	1	1/4	0
D008	1.00	1/1	1.00	N/A	1/3	0
D009	1.00	2/2	1.00	N/A	1/5	0
D010	1.00	1/1	1.00	1	1/3	0
D011	1.00	0/3	1.00	1	1/2	0
D012	1.00	3/4	0.50	1	2/5	0
D013	1.00	3/4	1.00	1	1/4	0
D014	1.00	3/3	1.00	1	1/3	0
D015	1.00	1/2	1.00	N/A	1	0
D016	1.00	2/2	1.00	1	1	0
D017	1.00	4/4	1.00	1	1/3	0
D018	1.00	3/4	1.00	1	1/3	0
D019	1.00	2/2	0.50	1	1/4	0
D020	1.00	0/2	1.00	1	2/5	0

Initially the 500 articles are downloaded from the ACM anthology [34]. Among the 500 articles, some articles are duplicated, some articles are not technical papers, for example, D379: DATABASE CONFERENCE CALENDAR (for details information on D379 see Appendix A). Some articles cannot be read at all due to converting problems, for example D312: An Aspect of Query Optimization in Multidatabase Systems: when opening the converted text file of D312, it is found that the whole file consists of non-readable characters. The schema of SH is designed for scientific paper (including abstract, keywords etc), CINDI system can use filters to remove the non-scientific papers from CINDI database successfully so the non-scientific papers are not counted in the test. Also, when applying ASHG to an article for the CINDI database, the article is tested first and if it was a non-readable article, the corresponding generated semantic header file and the text file would be removed. Thus we don't use non-readable articles for our tests. Therefore the total number of valid articles among the 500 was 452. For each test article, there would be a score for each field to be generated. For example, for D001, the ASHG was able to generate its title correctly and was noted as G = E, where G refers to the generated title and E refers to the title in the original article; hence the score for the title field will be 1. If the title generated by the ASHG is part of the title of the original article, G < E will be noted for the title field and the title score would be 0.5. If the title generated by the ASHG contains redundant string, then G > E will be noted for the title field and 0.5 will be assigned to the title score. In the author name field, if there were 3 authors in the article, but only 2 of the author names are generated correctly by the ASHG, then 2/3 will be the score for the author name field. Since score is used for abstract extraction and keywords extraction. If the abstract was not explicitly stated, the ASHG will try to get the

first paragraph of the introduction that would be used as the paper's abstract. If the keywords are not explicitly stated, the ASHG will try to generate at least one keyword for the article. Therefore, G will be noted in the keyword field. Since it is difficult to evaluate the accuracy of the generated keywords, the testing results for the keywords will only count on the explicitly stated keywords. So N/A will be marked in the field of keyword score. The rule for the subject is the same as the author name. However, as mentioned earlier, the subject classification generated by the ASHG is compared with Inspec's Classification Codes for the corresponding article.

Inspec is the leading English-language bibliographic database produced by the Institution of Electrical Engineers (IEE). It provides access to the worldwide literature on physics, electrical engineering and electronics, control theory and technology, and computers and computing. Its records contain bibliographic information, indexing terms, abstracts, property information, and element terms. In general, subject indexing and classification will only be seen in records from electronic services. We will compare our results with the electronic Inspec found at Concordia University's library.

Because the subject generated by the ASHG is not exactly the same as the Inspec, a subjective judgment has been made. Some articles listed in the table are not indexed by the Inspec, therefore N/A will be marked for both subject and subject score field.

5.1.1 Sample Results

In this section, we will show the complete semantic header generated by ASHG for two articles: D1 and D8 listed in Table 5.1.

The results of D1:

<semhdrB>
<useridB> <useridE>
<passwordB> <passwordE>

<titleB> Broadcast Protocols to Support Efficient Retrieval from Databases by Mobile Users <titleE>
<alttitleB> <alttitleE>
<subjectB>
<GeneralB> Computer Science <GeneralE>
<sublevel1B> Software <sublevel1E>
<sublevel2B> computer programs and softwares <sublevel2E>
<GeneralB> Computer Science <GeneralE>
<sublevel1B> Processor architectures <sublevel1E>
<sublevel2B> adaptable cellular architecture: mobile <sublevel2E>
<GeneralB> Electrical Engineering <GeneralE>
<sublevel1B> Communications <sublevel1E>
<sublevel2B> telecommunication applications <sublevel2E>
<GeneralB> Electrical Engineering <GeneralE>
<sublevel1B> Communications <sublevel1E>
<sublevel2B> communication: radio and television broadcasting <sublevel2E>
<GeneralB> Computer Science <GeneralE>
<sublevel1B> Processor architectures <sublevel1E>
<sublevel2B> parallel mobile processors <sublevel2E>
<GeneralB> Electrical Engineering <GeneralE>
<sublevel1B> Communications <sublevel1E>
<sublevel2B> microwave communication systems <sublevel2E>
<GeneralB> Electrical Engineering <GeneralE>
<sublevel1B> Communications <sublevel1E>
<sublevel2B> mobile communication systems <sublevel2E>
<GeneralB> Electrical Engineering <GeneralE>
<sublevel1B> Communications <sublevel1E>
<sublevel2B> wireless mobile communication systems <sublevel2E>
<GeneralB> Electrical Engineering <GeneralE>
<sublevel1B> Communications <sublevel1E>
<sublevel2B> telecommunication: mobile radio link systems <sublevel2E>
<GeneralB> Electrical Engineering <GeneralE>
<sublevel1B> Communications <sublevel1E>

<sublevel2B> information and communication protocols <sublevel2E>
<subjectE>
<languageB> English <languageE>
<char-setB> <char-setE>
<authorB>
<anameB> ANINDYA DATTA <anameE>
<anameB> DEBRA E. VANDERMEER <anameE>
<anameB> ASLIHAN CELIK <anameE>
<authorE>
<keywordB> Adaptive broadcast protocols, client-server computing, energy conservation, mobile databases <keywordE>
<identifierB>
<domain3B> FTP <domain3E>
<value3B> <value3E>
<identifierE>
<datesB>
<createdB> 2003/8/13 <createdE>
<expiryB> <expiryE>
<datesE>
<versionB> <versionE>
<spversionB> <spversionE>
<classificationB>
<domain4B> <domain4E>
<value4B> <value4E>
<classificationE>
<coverageB>
<domain5B> <domain5E>
<value5B> <value5E>
<coverageE>
<system-requirementsB>
<componentB> <componentE>
<exiganceB> <exiganceE>
<system-requirementsE>
<genreB>
<formB> <formE>
<sizeB> 175811 <sizeE>
<genreE>
<source-referenceB>
<relationB> <relationE>
<domain-identifierB> <domain-identifierE>
<source-referenceE>
<costB> <costE>
<abstractB>
Mobile computing has the potential for managing information globally. Data management issues in mobile computing have received some attention in recent times, and the design of adaptive broadcast protocols has been posed as an important problem.

Such protocols are employed by database servers to decide on the content of broadcasts dynamically, in response to client mobility and demand patterns. In this paper we design such protocols and also propose efficient retrieval strategies that may be employed by clients to download information from broadcasts. The goal is to design cooperative strategies between server and client to provide access to information in such a way as to minimize energy expenditure by clients. We evaluate the performance of our protocols both analytically and through simulation.
<abstractE>
<annotationB>
<annotationE>
<semhdrE>
<EOF>

The results of D8:

<semhdrB>
<useridB> <useridE>
<passwordB> <passwordE>
<titleB> Safe Query Languages for Constraint Databases <titleE>
<alttitleB> <alttitleE>
<subjectB>
<GeneralB> Computer Science <GeneralE>
<sublevel1B> Programming languages <sublevel1E>
<sublevel2B> computer program language <sublevel2E>
<GeneralB> Computer Science <GeneralE>
<sublevel1B> Software <sublevel1E>
<sublevel2B> computer programs and softwares <sublevel2E>
<GeneralB> Computer Science <GeneralE>
<sublevel1B> Information interfaces and presentation <sublevel1E>
<sublevel2B> user interfaces: input devices and strategies <sublevel2E>
<GeneralB> Computer Science <GeneralE>
<sublevel1B> Logics and meanings of programs <sublevel1E>
<sublevel2B> program type structure <sublevel2E>
<GeneralB> Computer Science <GeneralE>
<sublevel1B> Programming languages <sublevel1E>
<sublevel2B> constraint and logic languages <sublevel2E>
<GeneralB> Computer Science <GeneralE>
<sublevel1B> Programming languages <sublevel1E>
<sublevel2B> constraints in language constructs and features <sublevel2E>
<subjectE>
<languageB> English <languageE>
<char-setB> <char-setE>
<authorB>
<anameB> PETER Z. REVESZ <anameE>
<authorE>

<keywordB> query , constraint , use , type , symbol , program , paper , input , framework , form , close , approach <keywordE>
<identifierB>
<domain3B> FTP <domain3E>
<value3B> <value3E>
<identifierE>
<datesB>
<createdB> 2003/8/13 <createdE>
<expiryB> <expiryE>
<datesE>
<versionB> <versionE>
<spversionB> <spversionE>
<classificationB>
<domain4B> <domain4E>
<value4B> <value4E>
<classificationE>
<coverageB>
<domain5B> <domain5E>
<value5B> <value5E>
<coverageE>
<system-requirementsB>
<componentB> <componentE>
<exiganceB> <exiganceE>
<system-requirementsE>
<genreB>
<formB> <formE>
<sizeB> 107368 <sizeE>
<genreE>
<source-referenceB>
<relationB> <relationE>
<domain-identifierB> <domain-identifierE>
<source-referenceE>
<costB> <costE>
<abstractB>
In the database framework of Kanellakis et al. [1990] it was argued that constraint query Languages should take constraint databases as input and give other constraint databases that use the same type of atomic constraints as output. This closed-form requirement has been difficult to realize in constraint query languages that contain the negation symbol. This paper describes a general approach to restricting constraint query languages with negation to safe subsets that contain only programs that are evaluable in closed-form on any valid constraint database input.
<abstractE>
<annotationB>
<annotationE>
<semhdrE>
<EOF>

5.2 Results and Analysis

After conducting the tests for 452 articles, the statistics is based on the score of each field shown in the table in Appendix C. ASHG's average percentage accuracy for title, author name, abstract, keywords and subject classification is 92%, 80%, 92%, 93%, and 41% respectively. The test results and the figure are shown in the Table 5.3 and Figure 5.1.

Table 5.3 Test results

	Title	Author name	Abstract	Keywords	Subject classification
Average Accuracy (%)	92	80	92	93	41
Total Valid Articles	452				

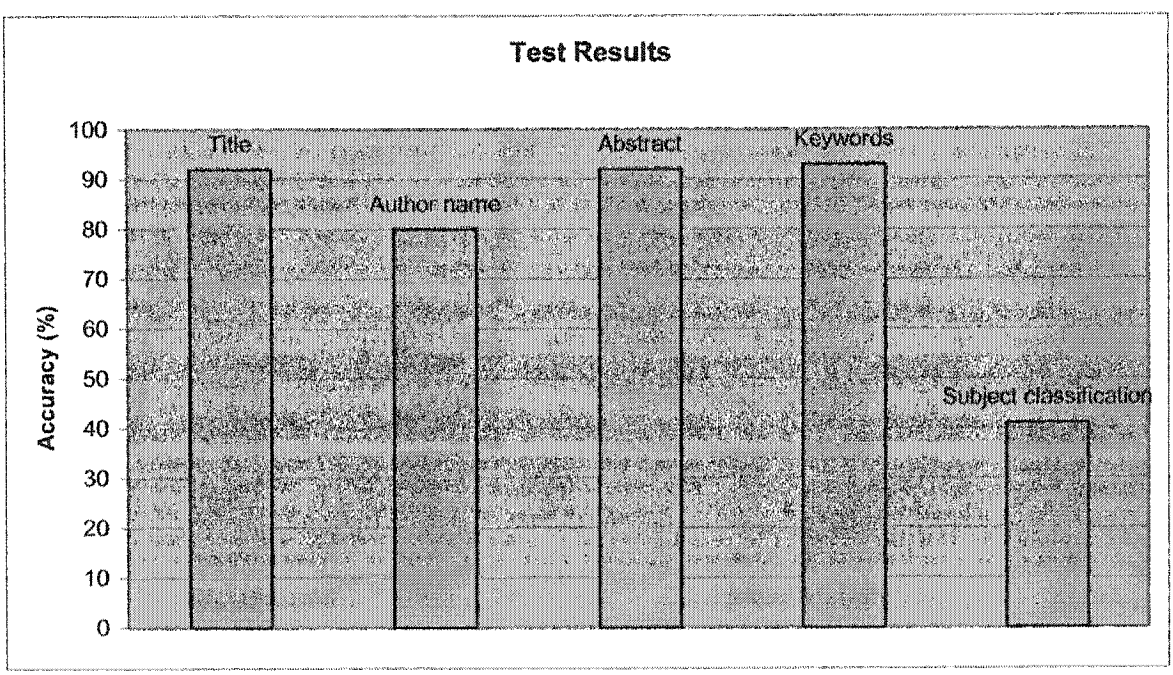

FIGURE 5.1 THE TESTING RESULTS

In addition, in order to determine the accuracy of converting PDF document to text file, an extra testing field for the 500 articles is added as "converting problems" shown in Table 5.1 and Appendix C. If in general there is no problems, "0" will be put in the field, otherwise "1" will be marked for the field. Among the 500 documents, only 24 articles cannot be converted correctly, 95% articles can be converted correctly. The results are shown below:

Table 5.4 The results of converting PDF document to Text file

	Number	Percentage
Non-readable paper	3	3/500= 0.6%
Paper with No space between words	21	21/500 = 4.2%
Total non-correctly converted paper	24	24/500 = 5 %
Total number of tested papers	500	
Average Accuracy		95%

From Table 5.2 and Figure 5.1, it is easy to see that the test results except subject classification are fairly high. The ASHG is able to extract most of the fields such as title, author name, abstract and explicit keywords accurately. However, subject classification is the weakest point of the ASHG. Although implicit keywords are not counted, it is worth to mention that ASHG generates a list of words for the implicit keywords that include some words that are insignificant. These insignificant words in turn lead to the diversion in subject classification [6]. Since Xpdf cannot convert some words correctly, it will also

decrease the efficiency of ASHG. For example, when Xpdf has problems to convert author name correctly, the proper author name cannot be generated since there is no match in the name database.

Chapter 6

Conclusion and Future work

6.1 Conclusion

Since there is large volume of information on the web, searching for specific information based on keywords has become imprecise. Thus, providing meta-information such as the Semantic Header for the information resource is necessary. The ASHG is a useful application, which automatically extracts and generates such meta-information with high accuracy for most of the usually required search terms. Since PDF document is portable, reliable and easy to use, it has become more popular. Consequently, the current project is to include PDF documents in the CINDI database, which would be located by another CINDI web robot project.

PDF document, using Adobe imaging model to represent text and graphics, consists of non-displayable characters due to. To analyze such documents we need to convert them to the text files, and then extract title, abstract, keyword and author name from the text file and construct the three level subject classifications. Xpdf is an open source software for the conversion, and after some modifications given good performance under Linux environment for our purpose. (The average accuracy of converting is about 95%). After applying the ASHG on 452 PDF documents, the final results show a significant accuracy on fields of Title, Abstract, Keywords and Author name in Semantic Header; the average

accuracy is about 92%, 92%, 93% and 80% respectively. However, it also show the weakest point of the ASHG is the subject classification, the average accuracy is about 41%.

6.2 Contribution of this thesis

The objective of this thesis is to generate semantic header from PDF document automatically. The design and implementation of the ASHG for PDF document is one of the main contributions made by this thesis to CINDI system; this includes the extraction of title, author name, email/phone number, abstract, the improvement of the syntax used in ASHG to govern the extraction of implicitly stated abstract, building the author name database and the web interface with the ASHG system and the CINDI system. In order to better format the converted text file, a modification to the original Xpdf was made. The design and implementation of the database to maintain and manage the generated semantic header files for downloaded articles from the Internet by the web robot also is another contribution of this thesis. By sharing database with the CINDI web robot and PDF converter, which converts non-PDF files (such as DOC format files) to PDF files, is a contribution to the CINDI system.

6.3 Future Work and Suggestions

Based on the test results, it is clear that the results for the subject classification should be improved. Since generating the subject hierarchy depends on the accuracy of the keywords extracted, the algorithm of generating keywords from the implicitly keywords

extraction should be improved. Currently, only single words are generated by ASHG; research into use of phrases may be useful in improving subject classification.

The ASHG can also generate semantic header from HTML document, Latex document, RTF document and Text document. The testing done by Haddad [6] and Zhang [35] were for only a limited number of documents for each type respectively; these types of documents can also be converted to PDF documents by the software. It is suggested that more documents of these four types should be tested, and then compared with the converted PDF documents respectively. The results will give a basic idea how to use the ASHG more efficiently.

Besides, the ASHG is also related to CINDI and ConfSys system, a conference management system, both under development in other projects. The integration of these three parts requires improvements. For example, for the web interface shown in Figure 2.3 in section 2.6, the button "Add One or More Subjects" should change to "Verify and Modify Subjects" since the accuracy of the subject classification is not high according to the experimental tests and the position of the button should move to the bottom of the web page.

Appendix A

The special character and corresponding HTML escape code

Special Character	HTML escape code
Æ	&Aelig;
Á	Á
Â	Â
À	À
Å	Å
Ã	Ã
Ä	Ä
Ç	Ç
Ð	Ð
É	É
Ê	Ê
È	È
Ë	Ë
Í	Í
Î	Î
Ì	Ì
Ï	Ï
Ñ	Ñ
Ó	Ó
Ô	Ô
Ò	Ò
Ø	Ø
Õ	Õ
Ö	Ö
Þ	Þ
Ú	Ú
Û	Û
Ù	Ù
Ü	Ü
Ý	Ý
á	á
â	â
æ	æ
à	à
a	α
å	å
ã	ã
ä	ä
ß	β

ç	ç
d	δ
é	é
ê	ê
è	è
e	ε
Ð	ð
ë	ë
í	í
î	î
ì	ì
ï	ï
µ	μ
ñ	ñ
ó	ó
ô	ô
ò	ò
ø	ø
õ	õ
ö	ö
f	φ
p	π
s	σ
ß	ß
t	τ
Þ	þ
ú	ú
û	û
ù	ù
ü	ü
ý	ý
Y	ÿ

Appendix B

Papers Used in Testing ASHG for PDF Document Format

PDF No.	Journal /Proceedings	Volume / Year	Testing Paper Title
D001	TODS	Vol 24, 1999	Broadcast Protocols to Support Efficient Retrieval from Databases by Mobile Users
D002	TODS	Vol 24, 1999	Database Design for Incomplete Relations
D003	TODS	Vol 24, 1999	Temporal FDs on Complex Objects
D004	TODS	Vol 24, 1999	Optimization of Queries with User-Defined Predicates
D005	TODS	Vol 24, 1999	GlOSS: Text-Source Discovery over the internet
D006	TODS	Vol 24, 1999	Distance Browsing in Spatial Databases
D007	TODS	Vol 24, 1999	Supporting Valid-Time Indeterminacy
D008	TODS	Vol 24, 1999	Safe Query Languages for Constraint Databases
D009	TODS	Vol 24, 1999	Safe Stratified Datalog With Integer Order Does Not Have Syntax
D010	TODS	Vol 24, 1999	Optimization Techniques for Queries with Expensive Methods
D011	TODS	Vol 23, 1998	Multiview Access Protocols for Large-Scale Replication
D012	TODS	Vol 23, 1998	Ensuring Consistency in Multidatabases by Preserving Two-Level Serializability
D013	TODS	Vol 23, 1998	An Access Control Model Supporting Periodicity Constraints and Temporal Reasoning
D014	TODS	Vol 23, 1998	Conceptual Schema Analysis: Techniques and Applications+N31
D015	TODS	Vol 23, 1998	An Efficient Method for Checking Object-Oriented Database Schema Correctness
D016	TODS	Vol 23, 1998	Information Gathering in the World Wide Web: The W3QL Query Language and the W3QS System
D017	TODS	Vol 23, 1998	Towards a Theory of Cost Management for Digital Libraries and Electronic Commerce
D018	TODS	Vol 23, 1998	Inverted Files Versus Signature Files for Text Indexing
D019	TODS	Vol 22, 1997	Extended Ephemeral Logging: Log Storage Management for Applications with Long-Lived Transactions
D020	TODS	Vol 22, 1997	Outerjoin Simplification and Reordering for Query Optimization
D021	TODS	Vol 22, 1997	An Axiomatic Model of Dynamic Schema Evolution in Objectbase Systems
D022	TODS	Vol 22, 1997	Logical Design for Temporal Databases with Multiple Granularities

D023	TODS	Vol 22, 1997	On the Semantics of "Now" in Databases
D024	TODS	Vol 22, 1997	Applying Formal Methods to Semantic-Based Decomposition of Transactions
D025	TODS	Vol 22, 1997	An Adaptive Data Replication Algorithm
D026	TODS	Vol 22, 1997	Transactional Client-Server Cache Consistency: Alternatives and Performance
D027	TODS	Vol 22, 1997	Disjunctive Datalog
D028	TODS	Vol 22, 1997	ProbView: A Flexible Probabilistic Database System
D029	TODS	Vol 22, 1997	Database Design with Common Sense Business Reasoning and Learning
D030	TODS	Vol 22, 1997	Object Normal Forms and Dependency Constraints for Object-Oriented Schemata
D031	TODS	Vol 22, 1997	Adaptive, Fine-Grained Sharing in a Client-Server OODBMS: A Callback-Based Approach
D032	TODS	vol 21, 1996	Modularization Techniques for Active Rules Design
D033	TODS	vol 21, 1996	Polymorphism and Type Inference in Database Programming
D034	TODS	vol 21, 1996	A Normal Form Redundancy infor Precisely Characterizing Nested Relations
D035	TODS	vol 21, 1996	Magic Conditions
D036	TODS	vol 21, 1996	The Building Blocks for Specifying Communication Behavior of Complex Objects: An Activity-Driven Approach
D037	TODS	vol 21, 1996	Tail Recursion Elimination in Deductive Databases
D038	TODS	vol 21, 1996	Implementing Deductive Databases by Mixed Integer Programming
D039	TODS	vol 21, 1996	Solving Satisfiability and Implication Problems in Database Systems
D040	TODS	vol 21, 1996	Declustering of Key-Based Partitioned Signature Files
D041	TODS	vol 21, 1996	A Probabilistic Relational Model and Algebra
D042	TODS	vol 21, 1996	Heraclitus: Elevating Deltas to be First- Class Citizens in a Language+N59
D043	TODS	vol 21, 1996	Model and Verification of a Data Manager Based on ARIES
D044	TODS	vol 21, 1996	LH*--A Scalable, Distributed Data Structure
D045	TODS	vol 21, 1996	Semantics for Update Rule Programs and Implementation in a Relational Database Management System
D046	The VLDB Journal	vol 6, 1997	The hB -tree: a multi-attribute index supporting concurrency, recovery and node consolidation
D047	The VLDB Journal	vol 6, 1997	Dictionary-based order-preserving string compression
D048	The VLDB Journal	vol 6, 1997	Analysis of locking behavior in three real database systems
D049	The VLDB Journal	vol 6, 1997	Data placement in shared-nothing parallel database systems
D050	The VLDB Journal	vol 6, 1997	A database model for object dynamics
D051	The VLDB Journal	vol 6, 1997	Graphical interaction with heterogeneous databases

D052	The VLDB Journal	vol 6, 1997	On applying hash filters to improving the execution of multi-join queries
D053	The VLDB Journal	vol 6, 1997	Parametric query optimization
D054	The VLDB Journal	vol 6, 1997	Concurrency control in hierarchical multidatabase systems
D055	The VLDB Journal	vol 6, 1997	The impact of object technology on commercial transaction processing
D056	The VLDB Journal	vol 6, 1997	Heuristic and randomized optimization for the join ordering problem
D057	The VLDB Journal	vol 6, 1997	Synchronization and recovery in a client-server storage system
D058	The VLDB Journal	vol 6, 1997	Concurrency and recovery for index trees
D059	The VLDB Journal	vol 6, 1997	SEEKing the truth about ad hoc join costs
D060	The VLDB Journal	vol 6, 1997	Erratum A database model for object dynamics
D061	The VLDB Journal	vol 6, 1997	Query processing over object views of relational data
D062	The VLDB Journal	vol 6, 1997	EXACT: an extensible approach to active object-oriented databases
D063	The VLDB Journal	vol 6, 1997	Structured document storage and refined declarative and navigational access mechanisms in HyperStorM
D064	The VLDB Journal	vol 6, 1997	A configurable type hierarchy index for OODB
D065	The VLDB Journal	vol 6, 1997	Using extended feature objects for partial similarity retrieval
D066	The VLDB Journal	vol 5, 1996	Parallelizing OODBMS traversals: a performance evaluation
D067	The VLDB Journal	vol 5, 1996	Priority assignment in real-time active databases
D068	The VLDB Journal	vol 5, 1996	A predicate-based caching scheme for client-server database architectures
D069	The VLDB Journal	vol 5, 1996	Mariposa: a wide-area distributed database system
D070	The VLDB Journal	vol 5, 1996	Join algorithm costs revisited
D071	The VLDB Journal	vol 5, 1996	A taxonomy of correctness criteria in database applications
D072	The VLDB Journal	vol 5, 1996	The GMAP: a versatile tool for physical data independence
D073	The VLDB Journal	vol 5, 1996	Algebraic query optimisation for database programming languages
D074	The VLDB Journal	vol 5, 1996	Type-safe relaxing of schema consistency rules for flexible modelling in OODBMS
D075	The VLDB Journal	vol 5, 1996	An experimental object-based sharing system for networked databases
D076	The VLDB Journal	vol 5, 1996	A complete temporal relational algebra
D077	The VLDB Journal	vol 5, 1996	The design and implementation of K: a high-level knowledge-base programming language of OSAM*.KBMS
D078	The VLDB Journal	vol 5, 1996	Access path support for referential integrity in SQL2
D079	The VLDB Journal	vol 5, 1996	Index nesting an efficient approach to indexing in object-oriented databases
D080	The VLDB Journal	vol 5, 1996	Query processing and optimization in Oracle Rdb
D081	The VLDB Journal	vol 5, 1996	Building knowledge base management systems
D082	The VLDB Journal	vol 5, 1996	An asymptotically optimal multiversion B-tree

D083	The VLDB Journal	vol 5, 1996	Semantic and schematic similarities between database objects: a context-based approach
D084	The VLDB Journal	vol 4, 1995	The Software Information Base: A Server for Reuse
D085	The VLDB Journal	vol 4, 1995	HyperFile: A Data and Query Model for Documents
D086	The VLDB Journal	vol 4, 1995	Ordered Shared Locks for Real-Time Databases
D087	The VLDB Journal	vol 4, 1995	Characterization of Database Access Pattern for Analytic Prediction of Buffer Hit Probability
D088	The VLDB Journal	vol 4, 1995	Data Model for Extensible Support of Explicit Relationships in Design Databases
D089	The VLDB Journal	vol 4, 1995	Updating Knowledge Bases While Maintaining Their Consistency
D090	The VLDB Journal	vol 4, 1995	Realm-Based Spatial Data Types: The ROSE Algebra
D091	The VLDB Journal	vol 4, 1995	InterViso: Dealing With the Complexity of Federated Database Access
D092	The VLDB Journal	vol 4, 1995	Orthogonally PersistentObject Systems
D093			Not exist
D094	The VLDB Journal	vol 4, 1995	TIGUKAT: A Uniform Behavioral Objectbase Management System
D095	The VLDB Journal	vol 4, 1995	Thémis: A Database Programming Language Handling Integrity Constraints.
D096	The VLDB Journal	vol 4, 1995	Thémis: A Database Programming Language Handling Integrity Constraints.
D097	The VLDB Journal	vol 4, 1995	Adaptable Pointer Swizzling Strategies in Object Bases: Design, Realization, and Quantitative Analysis
D098	The VLDB Journal	vol 4, 1995	Sleepers and Workaholics: Caching Strategies in Mobile Environments (Extended Version)
D099	The VLDB Journal	vol 4, 1995	AlphaSort: A Cache-Sensitive Parallel External Sort
D100	The VLDB Journal	vol 4, 1995	Estimating Page Fetches for Index Scans with Finite LRU Buffers
D101	The VLDB Journal	vol 4, 1995	Historical Queries Along Multiple Lines of Time Evolution
D102	The VLDB Journal	vol 4, 1995	The Power of Languages for the Manipulation of Complex Values
D103	The VLDB Journal	vol 4, 1995	Chronological Scheduling of Transactions with Temporal Dependencies
D104	The VLDB Journal	vol 4, 1995	Dynamic Maintenance of Data Distribution for Selectivity Estimation
D105	The VLDB Journal	vol 4, 1995	A Pattern-Based Object Calculus
D106	The VLDB Journal	vol 4, 1995	Versioning and Configuration Management in an Object-Oriented Data Model
D107	The VLDB Journal	vol 4, 1995	An Introduction to Deductive Database Languages and Systems
D108	The VLDB Journal	vol 4, 1995	The Glue-Nail Deductive Database System: Design, Implementation, and Evaluation
D109	The VLDB Journal	vol 4, 1995	The CORAL Deductive System
D110	The VLDB Journal	vol 4, 1995	DECLARE and SDS: Early Efforts to Commercialize Deductive Database Technology

D111	The VLDB Journal	vol 4, 1995	The Aditi Deductive Database System
D112	The VLDB Journal	vol 4, 1995	The Demarcation Protocol: A Technique for Maintaining Constraints in Distributed Database Systems
D113	The VLDB Journal	vol 4, 1995	Index Configuration in Object-Oriented Databases
D114	The VLDB Journal	vol 4, 1995	An Introduction to Spatial Database Systems
D115	The VLDB Journal	vol 4, 1995	Management of Multidimensional Discrete Data
D116	The VLDB Journal	vol 4, 1995	A Semantic Modeling Approach for Image Retrieval by Content
D117	The VLDB Journal	vol 4, 1995	Qualitative Representation of Spatial Knowledge in Two-Dimensional Space
D118	The VLDB Journal	vol 4, 1995	The W-Tree: An Index Structure for High-Dimensional Data
D119	The VLDB Journal	vol 2, 1993	Buffer Management Based on Return on Consumption In a Multi-Query Environment
D120	The VLDB Journal	vol 2, 1993	Concurrency Control Issues in Nested Transactions
D121	The VLDB Journal	vol 2, 1993	Using Differential Techniques to Efficiently Support Transaction Time
D122	The VLDB Journal	vol 2, 1993	Value-Based Scheduling in Real-Time Database Systems
D123	The VLDB Journal	vol 2, 1993	Query Languages for Relational Multidatabases
D124	The VLDB Journal	vol 2, 1993	Generating Consistent Test Data: Restricting the Search Space by a Generator Formula
D125	The VLDB Journal	vol 2, 1993	Supporting Consistent Updates in Replicated Multidatabase Systems
D126	The VLDB Journal	vol 2, 1993	Query Processing and Inverted Indices in Shared-Nothing Text Document Information Retrieval Systems+O138
D127	The VLDB Journal	vol 2, 1993	Multi-Level Transaction Management for Complex Objects: Implementation, Performance, Parallelism
D128	The VLDB Journal	vol 2, 1993	Understanding Semantic Relationships
D129	The VLDB Journal	vol 2, 1993	Searching a Minimal Semantically-Equivalent Subset of a Set of Partial Values
D130	The VLDB Journal	vol 2, 1993	Parallel Query Processing With Zigzag Trees
D131	The VLDB Journal	vol 2, 1993	Considering Data Skew Factor in Multi-Way Join Query Optimization for Parallel Execution
D132	The VLDB Journal	vol 2, 1993	A Theory of Global Concurrency Control in Multidatabase Systems
D133	The VLDB Journal	vol 1. 1992	Transaction Management Issues in a Failure-Prone Multidatabase System Environment
D134	The VLDB Journal	vol 1. 1993	Cooperative Transaction Hierarchies: Transaction Support for DesignApplications
D135	The VLDB Journal	vol 1. 1994	Model Independent Assertions for Integration of Heterogeneous Schemas
D136	The VLDB Journal	vol 1. 1995	Federated Databases and Systems: Part I - A Tutorial on Their Data Sharing
D137	The VLDB Journal	vol 1. 1996	Overview of Multidatabase Transaction Management
D138	The VLDB Journal	vol 1. 1997	A Toolkit for the Incremental Implementation of Heterogeneous Database Management Systems

D139	The VLDB Journal	vol 1. 1998	Federated Databases and Systems: Part II - A Tutorial on Their Resource Consolidation
D140	The VLDB Journal	vol 1. 1999	SIGMOD Sister Societies
D141	SIGMOD	vol 29. 2000	ACM-SIGMOD Digital Review:Restaurant Ratings for Technical Papers
D142	SIGMOD	vol 29, 2000	Evolution and Change in Data Management - Issues and Directions
D143	SIGMOD	vol 29, 2000	Generating dynamic content at database-backed web servers" cgi-bin vs mod_perl
D144	SIGMOD	vol 29, 2000	Hierarchies and Relative Operators in the OLAP Environment
D145	SIGMOD	vol 29, 2000	An Optimisation Scheme for Coalesce/Valid Time Selection Operator Sequences
D146	SIGMOD	vol 29, 2000	Incremental Maintenance of Recursive Views Using Relational Calculus/SQL*
D147	SIGMOD	vol 29, 2000	Reminiscences on Influential Papers
D148	SIGMOD	vol 29, 2000	An Extensible Compressor for XML Data
D149	SIGMOD	vol 29, 2000	SQL Standardization: The Next Steps
D150	SIGMOD	vol 29, 2000	Comparative Analysis of Five XML Query Languages
D151	SIGMOD	vol 29, 2000	Beyond Document Similarity: Understanding Value-Based Search and Browsing Technologies
D152	SIGMOD	vol 29, 2000	Workshop on Performance and Architecture of Web Servers(PAWS-2000)
D153	SIGMOD	vol 29, 2000	Report on ISDO '00: The CAiSE*00 Workshop on "Infrastructures for Dynamic Business-to-Business Service Outsourcing
D154	SIGMOD	vol 29, 2000	Report on Second International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems
D155	SIGMOD	vol 29, 2000	Provision of Market Services for eCo Compliant Electronic Marketplaces
D156	SIGMOD	vol 29, 2000	Spatial Operators
D157	SIGMOD	vol 29, 2000	Generating Spatiotemporal Datasets on the WWW
D158	SIGMOD	vol 29, 2000	Constraint databases: A tutorial introduction
D159	SIGMOD	vol 29, 2000	
D160	SIGMOD	vol 29, 2000	The Implementation and Performance of Compressed Databases
D161	SIGMOD	vol 29, 2000	
D162	SIGMOD	vol 29, 2000	Comparative Analysis of Six XML Schema Languages
D163	SIGMOD	vol 29, 2000	Knowledge Discovery in Data Warehouses
D164	SIGMOD	vol 28, 1999	Semantic Interoperability in Global Information Systems A brief introduction to the research area and the special section
D165	SIGMOD	vol 28, 1999	Semantic Integration of Environmental Models for Application to Global Information Systems and Decision-Making

D166	SIGMOD	vol 28, 1999	Semantic and Pedagogic Interoperability Mechanisms in the ARIADNE Educational Repository
D167	SIGMOD	vol 28, 1999	Unpacking The Semantics of Source and Usage To Perform Semantic Reconciliation In Large-Scale Information Systems
D168	SIGMOD	vol 28, 1999	Semantic Video Indexing: Approach and Issues
D169	SIGMOD	vol 28, 1999	Contextualizing the Information Space in Federated Digital Libraries
D170	SIGMOD	vol 28, 1999	Dynamic Service Matchmaking Among Agents in Open.Information Environments*
D171	SIGMOD	vol 28, 1999	Semantic Integration of Semistructured and Structured Data Sources
D172	SIGMOD	vol 28, 1999	Agent-Based Semantic Interoperability in InfoSleuth
D173	SIGMOD	vol 28, 1999	Semantic Interoperability in Information Services: Experiencing with CoopWARE
D174	SIGMOD	vol 28, 1999	Design Principles for Data-Intensive Web Sites
D175	SIGMOD	vol 28, 1999	A study on data point search for HG-trees
D176	SIGMOD	vol 28, 1999	The OASIS Multidatabase Prototype
D177	SIGMOD	vol 28, 1999	VideoAnywhere:A System for Searching and Managing Distributed Heterogeneous Video Assets
D178	SIGMOD	vol 28, 1999	Reminiscences on Influential Papers
D179	SIGMOD	vol 28, 1999	NSF Workshop on Industrial/Academic Cooperation in Database Systems
D180	SIGMOD	vol 28, 1999	SQL:1999, formerly known as SQL3
D181	SIGMOD	vol 28, 1999	Towards Adaptive Workflow Systems
D182	SIGMOD	vol 28, 1999	Engineering Federated Information Systems Report of EFIS '99 Workshop
D183	SIGMOD	vol 28, 1999	Chorochronos: A Research Network for Spatiotemporal Database Systems
D184	SIGMOD	vol 28, 1999	Cost Estimation of User-Defined Methods in Object-Relational Database Systems
D185	SIGMOD	vol 28, 1999	First-Class Views:A Key to User-Centered Computing
D186	SIGMOD	vol 28, 1999	On Multi-resolution Document Transmission in Mobile Web
D187	SIGMOD	vol 28, 1999	Reminiscences on Influential Papers
D188	SIGMOD	vol 28, 1999	Distributed Transactions in Practice
D189	SIGMOD	vol 28, 1999	A Distributed Scientific Data Archive Using the Web, XML and SQL/MED
D190	SIGMOD	vol 28, 1999	An Overviewand Classification of Mediated Query Systems
D191	SIGMOD	vol 28, 1999	Database Research at The University of Oklahoma
D192	SIGMOD	vol 28, 1999	Timer-Driven Database Triggers and Alerters: Semantics and a Challenge
D193	SIGMOD	vol 28, 1999	Diluting ACID
D194	SIGMOD	vol 28, 1999	Some Remarks on Variable Independence, Closure, and Orthographic Dimension in Constraint Databases

D195	SIGMOD	vol 28, 1999	Database Principles Database Principles Column
D196	SIGMOD	vol 28, 1999	On Views and XML
D197	SIGMOD	vol 28, 1999	Reminiscences on Influential Papers
D198	SIGMOD	vol 28, 1999	FinTime - a financial time series benchmark
D199	SIGMOD	vol 28, 1999	Practical Lessons in Supporting Large-Scale Computational Science
D200	SIGMOD	vol 28, 1999	SQLJ Part 1: SQL Routines using the JavaTM Programming Language
D201	SIGMOD	vol 28, 1999	A Survey of Logical Models for OLAP Databases
D202	SIGMOD	vol 27, 1998	PREDATOR:A Resource for Database Research
D203	SIGMOD	vol 27, 1998	Materialized Views and Data Warehouses
D204	SIGMOD	vol 27, 1998	Applications of JAVA programming language to database management
D205	SIGMOD	vol 27, 1998	Unbundling Active Functionality
D206	SIGMOD	vol 27, 1998	Mining Fuzzy Association Rules in Databases
D207	SIGMOD	vol 27, 1998	no Title
D208	SIGMOD	vol 27, 1998	Not exist
D209	SIGMOD	vol 27, 1998	T2:A Customizable Parallel Database For Multi-dimensional Data
D210	SIGMOD	vol 27, 1998	Workow History Management
D211	SIGMOD	vol 27, 1998	Towards a Richer Web Object Model
D212	SIGMOD	vol 27, 1998	Where Will Object Technology Drive Data Administration?
D213	SIGMOD	vol 27, 1998	Repositories and Object Oriented Databases
D214	SIGMOD	vol 27, 1998	Towards On-Line Analytical Miningin Large Databases
D215	SIGMOD	vol 27, 1998	Enhanced Nearest Neighbour Search on the R-tree
D216	SIGMOD	vol 27, 1998	Algebraic Change Propagation for Semijoin and Outerjoin Queries
D217	SIGMOD	vol 27, 1998	no Title
D218	SIGMOD	vol 27, 1998	Reminiscences on Influential Papers
D219	SIGMOD	vol 27, 1998	The TriGS Active Object-Oriented Database System - An Overview
D220	SIGMOD	vol 27, 1998	A Case for Intelligent Disks (IDISKs)
D221	SIGMOD	vol 27, 1998	Standards In Practice
D222	SIGMOD	vol 27, 1998	Database Techniques for the World-Wide Web: A Survey
D223	SIGMOD	vol 27, 1998	DATABASE RESEARCH AT COLUMBIA UNIVERSITY
D224	SIGMOD	vol 27, 1998	The Microsoft Database Research Group
D225	SIGMOD	vol 27, 1998	Guest Editor's Introduction
D226	SIGMOD	vol 27, 1998	Component-based E-Commerce: Assessment of Current Practices and Future Directions
D227	SIGMOD	vol 27, 1998	Building Database-driven Electronic Catalogs
D228	SIGMOD	vol 27, 1998	XML and Electronic Commerce: Enabling the Network Economy
D229	SIGMOD	vol 27, 1998	A Workflow-based Electronic Marketplace on the Web

D230	SIGMOD	vol 27, 1998	ADEPT: An Agent-Based Approach to Business Process Management
D231	SIGMOD	vol 27, 1998	A Componentized Architecture for Dynamic Electronic Markets
D232	SIGMOD	vol 27, 1998	Design and Implementation of RMP - A Virtual Electronic Market Place
D233	SIGMOD	vol 27, 1998	Discovering Internet Marketing Intelligence through Online Analytical Web Usage Mining
D234	SIGMOD	vol 27, 1998	An Anonymous Electronic Commerce Scheme with an Off-Line Authority and Untrusted Agents
D235	SIGMOD	vol 27, 1998	Electronic market:The roadmap for university libraries and members to survive in the information jungle
D236	SIGMOD	vol 27, 1998	Reminiscences on Influential Papers
D237	SIGMOD	vol 27, 1998	The Middleware Muddle Application servers and TP monitors are finding new life on the Net.
D238	SIGMOD	vol 27, 1998	SQLJ Part O, now known as SQL/OLB
D239	SIGMOD	vol 26, 1997	Integrating Modelling Systems for Environmental Management Information Systems
D240	SIGMOD	vol 26, 1997	Improving Access to Environmental Data using Context Information
D241	SIGMOD	vol 26, 1997	WWW-UDK:A Web-based Environmental Meta-Information System
D242	SIGMOD	vol 26, 1997	Data Management for Earth System Science
D243	SIGMOD	vol 26, 1997	Open GIS and On-Line Environmental Libraries
D244	SIGMOD	vol 26, 1997	Mediator Languages
D245	SIGMOD	vol 26, 1997	A Consumer Viewpoint on "Mediator Languages a Proposal for a Standard"
D246	SIGMOD	vol 26, 1997	MIN-MAX COMPRESSION METHODSFOR MEDICAL IMAGE DATABASES
D247	SIGMOD	vol 26, 1997	A TSQL2 Tutorial
D248	SIGMOD	vol 26, 1997	Converting Relational to Object-Oriented Databases
D249	SIGMOD	vol 26, 1997	Extraction of Object-Oriented Structures from Existing Relational Databases
D250	SIGMOD	vol 26, 1997	no Title
D251	SIGMOD	vol 26, 1997	Query Previews for Networked Information Systems: A Case Study with NASA Environmental Data
D252	SIGMOD	vol 26, 1997	Opportunities in Information Management and Assurance
D253	SIGMOD	vol 26, 1997	A Query Language for a Web-Site
D254	SIGMOD	vol 26, 1997	Quasi-Cubes: Exploiting approximations in multidimensional databases
D255	SIGMOD	vol 26, 1997	OGDI: Toward Interoperability among Geospatial Databases
D256	SIGMOD	vol 26, 1997	An Extended Entity-Relationship Model for Geographic Applications
D257	SIGMOD	vol 26, 1997	Asserting Beliefs in MLS Relational Models
D258	SIGMOD	vol 26, 1997	Asserting Beliefs in MLS Relational Models
D259	SIGMOD	vol 26, 1997	Database Systems Breaking Out of the Box

D260	SIGMOD	vol 26, 1997	Lore: A Database Management System for Semistructured Data
D261	SIGMOD	vol 26, 1997	Research in Databases and Data-Intensive Applications
D262	SIGMOD	vol 26, 1997	Intelligent Access to Heterogeneous Information Sources Report on the 4th Workshop on Knowledge Representation Meets Databases
D263	SIGMOD	vol 26, 1997	Virtual Database Technology
D264	SIGMOD	vol 26, 1997	Industry Perspectives
D265	SIGMOD	vol 26, 1997	The Five-Minute Rule Ten Years Later,and Other Computer Storage Rules of Thumb
D266	SIGMOD	vol 26, 1997	INFORMATION SYSTEMS RESEARCH AT GEORGE MASON UNIVERSITY
D267	SIGMOD	vol 26, 1997	The Database and Information System Research Group at the University of Ulm
D268	SIGMOD	vol 25, 1996	Advances in Real-Time Database Systems Research
D269	SIGMOD	vol 25, 1996	Integrating Temporal, Real-Time, and Active Databases
D270	SIGMOD	vol 25, 1996	Reai-Time Index Concurrency Control
D271	SIGMOD	vol 25, 1996	Real-Time Database -- Similarity Semantics and Resource Scheduling
D272	SIGMOD	vol 25, 1996	Exploiting Main Memory DBMS Features to Improve Real-Time Concurrency Control Protocols
D273	SIGMOD	vol 25, 1996	Enhancing External Consistency in Real-Time Transactions
D274	SIGMOD	vol 25, 1996	Improving Timeliness in Real-Time Secure Database Systems
D275	SIGMOD	vol 25, 1996	Overview of tlhe STanford Real-time Information Processor (STRIP)
D276	SIGMOD	vol 25, 1996	DeeDS Towards a Distributed and Active Real-Time Database System
D277	SIGMOD	vol 25, 1996	no Title
D278	SIGMOD	vol 25, 1996	Integrating Contents and Structure in Text Retrieval
D279	SIGMOD	vol 25, 1996	Lifestreams: A Storage Model for Personal Data
D280	SIGMOD	vol 25, 1996	Object Query Standards
D281	SIGMOD	vol 25, 1996	DOMAINS, RELATIONS AND RELIGIOUS WARS
D282	SIGMOD	vol 25, 1996	Guidelines for Presentation and Comparison of Indexing Techniques
D283	SIGMOD	vol 25, 1996	Much Ado About Shared-Nothing
D284	SIGMOD	vol 25, 1996	On the Cost of Monitoring and Reorganization of Object Bases for Clustering
D285	SIGMOD	vol 25, 1996	Open Issuesin Parallel Query Optimization
D286	SIGMOD	vol 25, 1996	Control strategies for complex relational query processing in shared nothing systems
D287	SIGMOD	vol 25, 1996	The BeSS Object Storage Manager: Architecture Overview
D288	SIGMOD	vol 25, 1996	The Aggregate Data Problem:a System for their Definition and Management

D289	SIGMOD	vol 25, 1996	INFORMATION VISUALIZATION
D290	SIGMOD	vol 25, 1996	Dynamic Information Visualization
D291	SIGMOD	vol 25, 1996	Incremental Data Structures and Algorithms for Dynamic Query Interfaces
D292	SIGMOD	vol 25, 1996	Spotfire: An Information Exploration Environment
D293	SIGMOD	vol 25, 1996	A Framework for Information Visualisation
D294	SIGMOD	vol 25, 1996	Pixel-oriented Database Visualizations
D295	SIGMOD	vol 25, 1996	To Table or Not to Table: a Hypertabular Answer
D296	SIGMOD	vol 25, 1996	Applying Database Visualization to the World Wide Web
D297	SIGMOD	vol 25, 1996	3D Geographic Network Displays
D298	SIGMOD	vol 25, 1996	An Orthogonally Persistent Java
D299	SIGMOD	vol 25, 1996	The Mariposa Distributed Database Management System
D300	SIGMOD	vol 25, 1996	New Standard for Stored Procedures in
D301	SIGMOD	vol 24, 1995	Application ofOODB and SGML Techniques in Text Database: An Electronic Dictionary System
D302	SIGMOD	vol 24, 1995	Implementation Aspects of an Object-Oriented DBMS
D303	SIGMOD	vol 24, 1995	HODFA: An Architectural Framework for Homogenizing
D304	SIGMOD	vol 24, 1995	A Close Look at the IFO Data Model
D305	SIGMOD	vol 24, 1995	MULTIGRANULARITY LOCKING IN MULTIPLE JOB CLASSES TRANSACTION PROCESSING SYSTEM
D306	SIGMOD	vol 24, 1995	Implementing Deletion in B+-Trees
D307	SIGMOD	vol 24, 1995	The Third Manifesto
D308	SIGMOD	vol 24, 1995	An Annotated Bibliography of Benchmarks for Object Databases
D309	SIGMOD	vol 24, 1995	An Annotated Bibliography on Active Databases
D310	SIGMOD	vol 24, 1995	DESIGN AND USER TESTING OF A MULTI-PARADIGM QUERY INTERFACE TO AN OBJECT-ORIENTED DATABASE
D311	SIGMOD	vol 24, 1995	Mapping Extended Entity Relationship Model to Object Modeling Technique
D312	SIGMOD	vol 24, 1995	An Aspect of Query Optimizationin Multidatabase Systems
D313	SIGMOD	vol 24, 1995	An Introduction to Remy's Fast Polymorphic Record Projection
D314	SIGMOD	vol 24, 1995	On the Issue of Valid Time(s) in Temporal Databases
D315	SIGMOD	vol 24, 1995	A Framework for Providing Consistent and Recoverable Agent-Based Access to Heterogeneous Mobile Databases
D316	SIGMOD	vol 24, 1995	METU interoperable Database System
D317	SIGMOD	vol 24, 1995	Information Finding in a Digital Library: the Stanford Perspective
D318	SIGMOD	vol 24, 1995	
D319	SIGMOD	vol 24, 1995	Condition Handling in SQL Persistent Stored Modules

D320	SIGMOD	vol 24, 1995	From the Guest Editors
D321	SIGMOD	vol 24, 1995	MOBILE COMPUTING and DATABASES: ANYTHING NEW?
D322	SIGMOD	vol 24, 1995	A Research Status Report onAdaptation for Mobile Data Access
D323	SIGMOD	vol 24, 1995	Wireless Client/Server Computingfor Personal Information Services and Applications
D324	SIGMOD	vol 24, 1995	View Maintenance in Mobile Computing
D325	SIGMOD	vol 24, 1995	Managing Video Data in a Mobile Environment
D326	SIGMOD	vol 24, 1995	Digital Library Services in Mobile Computing
D327	SIGMOD	vol 24, 1995	An Annotated Bibliography on Real-Time Database Systems
D328	SIGMOD	vol 24, 1995	Parallelism and its Price : A Case Study of NonStop SQL/MP
D329	SIGMOD	vol 23, 1994	Research Perspectives for Time Series Management Systems
D330	SIGMOD	vol 23, 1994	A Hypertext Query Language for Images
D331	SIGMOD	vol 23, 1994	Supporting Dynmic Displays Using Active Rules
D332	SIGMOD	vol 23, 1994	Constructing the Next 100 Database Management Systems: Like the Handyman or Like the Engineer?
D333	SIGMOD	vol 23, 1994	Overview of the Special Sectionon Temporal Database Infrastructure
D334	SIGMOD	vol 23, 1994	Towards an Infrastructure for Temporal Databases: Report of an Invitational ARPA/NSF Workshop
D335	SIGMOD	vol 23, 1994	A Consensus Glossary of Temporal Database Concepts
D336	SIGMOD	vol 23, 1994	TSQL2 Language Specification
D337	SIGMOD	vol 23, 1994	Comprehension Syntax
D338	SIGMOD	vol 23, 1994	Text Databases: A Survey of Text Models and Systems
D339	SIGMOD	vol 23, 1994	Databases for GIS
D340	SIGMOD	vol 23, 1994	The Database Research Group at ETH Zurich
D341	SIGMOD	vol 23, 1994	Research Issues in Databases for ARCS: Active Rapidly Changing data Systems
D342	SIGMOD	vol 23, 1994	Influencing Database Language Standards
D343	SIGMOD	vol 23, 1994	Trade Press News
D344	SIGMOD	vol 23, 1994	Progress on HPCC and NII
D345	SIGMOD	vol 23, 1994	How to Modify SQL Queries in Order to Guarantee Sure Answers
D346	SIGMOD	vol 23, 1994	Performance Evaluation of A New Distributed Deadlock Detection Algorithm
D347	SIGMOD	vol 23, 1994	A TSQL2 Tutorial
D348	SIGMOD	vol 23, 1994	Research Issues in Active Database Systems:Report from the Closing Panel at RIDE-ADS '94
D349	SIGMOD	vol 23, 1994	Medical Information Systems: Characterization and Challenges
D350	SIGMOD	vol 23, 1994	Database Research at NTHU and ITRI
D351	SIGMOD	vol 23, 1994	Trade Press News

D352	SIGMOD	vol 23, 1994	Are the Terms "Version" and "Variant" Orthogonal to One Another? - A Critical Assessment of the STEP Standardization -
D353	SIGMOD	vol 23, 1994	Data Modelling in the Large
D354	SIGMOD	vol 23, 1994	A New Join Algorithm
D355	SIGMOD	vol 23, 1994	METADATA FOR DIGITAL MEDIA: INTRODUCTION TO THE SPECIAL ISSUE
D356	SIGMOD	vol 23, 1994	Metadata for Multimedia Documents
D357	SIGMOD	vol 23, 1994	Metadata in Video Databases
D358	SIGMOD	vol 23, 1994	A Metadatabase System for Semantic Image Search by a Mathematical Model of Meaning
D359	SIGMOD	vol 23, 1994	SEQUOIA 2000 METADATA SCHEMA FOR SATELLITE IMAGES
D360	SIGMOD	vol 23, 1994	Using Metadata for the Intelligent Browsing of Structured Media Objects
D361	SIGMOD	vol 23, 1994	METADATA FOR INTEGRATING SPEECH DOCUMENTS IN A TEXT RETRIEVAL SYSTEM
D362	SIGMOD	vol 23, 1994	METADATA FOR MIXED-MEDIA ACCESS
D363	SIGMOD	vol 23, 1994	Loading Databases Using Dataflow Parallelism
D364	SIGMOD	vol 23, 1994	Recent Design Trade-offs in SQL3
D365	SIGMOD	vol 22, 1993	not exist
D366	SIGMOD	vol 22, 1993	PARDES- A Data-Driven Oriented Active Database Model
D367	SIGMOD	vol 22, 1993	Parametric databases:seamless integration of spatial, temporal, belief and ordinary data
D368	SIGMOD	vol 22, 1993	Schema Transformation without Database Reorganization
D369	SIGMOD	vol 22, 1993	SIRIO: A DISTRIBUTED INFORMATION SYSTEM OVER A HETEROGENEOUS COMPUTER NETWORK
D370	SIGMOD	vol 22, 1993	Data Management for Mobile Computing
D371	SIGMOD	vol 22, 1993	WORKSHOP REPORTInternational Workshop on Distributed Object Management
D372	SIGMOD	vol 22, 1993	Remarks on two new theorems of Date and Fagin
D373	SIGMOD	vol 22, 1993	Response to "Remarks on two new theorems of Date and Fagin"
D374	SIGMOD	vol 22, 1993	Bibliography on Spatiotemporal Databases
D375	SIGMOD	vol 22, 1993	Extending the Scope of Database Services
D376	SIGMOD	vol 22, 1993	Database Research at Wisconsin
D377	SIGMOD	vol 22, 1993	Database Research at AT&T Bell Laboratories
D378	SIGMOD	vol 22, 1993	Change at ONR, and Many Funding Announcements Elsewhere
D379	SIGMOD	vol 22, 1993	DATABASE CONFERENCE CALENDAR
D380	SIGMOD	vol 22, 1993	Helping Computer Scientists in Romania
D381	SIGMOD	vol 22, 1993	On Temporal Modeling in the Context of Object Databases
D382	SIGMOD	vol 22, 1993	Schema Evolution in OODBs Using Class Versioning

D383	SIGMOD	vol 22, 1993	Merging Application-centricand Data-centric Approaches to Support Transaction-orientedMulti-system Workflows
D384	SIGMOD	vol 22, 1993	Database Compression
D385	SIGMOD	vol 22, 1993	DEADLOCK PREVENTION IN A DISTRIBUTED DATABASE SYSTEM
D386	SIGMOD	vol 22, 1993	Report of the Workshop on Semantic Heterogeneity and Interoperation in Multidatabase Systems
D387	PODS	1998	Dynamic Tree Isomorphism via First-order Updates to a Relational Database
D388	SIGMOD	vol 22, 1993	Options in Physical Database Design
D389	SIGMOD	vol 22, 1993	Database Research at the University of Florida
D390	SIGMOD	vol 22, 1993	Database Research at the University of Queensland
D391	SIGMOD	vol 22, 1993	Timely Access to Future Funding Announcements
D392	SIGMOD	vol 22, 1993	Relational Database Integration in the IBM AS/400
D393	SIGMOD	vol 22, 1993	MoodView: An Advanced Graphical User Interface for OODBMSs
D394	SIGMOD	vol 22, 1993	Experiences with HyperBase: A Hypertext Database Supporting Collaborative Work
D395	SIGMOD	vol 22, 1993	Implementation offa Graph-Based Data Model for Complex Objects
D396	SIGMOD	vol 22, 1993	Parallel Query Processing in Shared Disk Database Systems
D397	SIGMOD	vol 22, 1993	A PERFORMANCE STUDY OF CONCURRENCY CONTROL IN A REAL-TIME MAIN MEMORY DATABASE SYSTEM
D398	SIGMOD	vol 22, 1993	Role-Based Databases Security, Object Oriented & Separation of Duty
D399	SIGMOD	vol 22, 1993	CONCURRENCY CONTROL INTRUSTED DATABASE MANAGEMENT SYSTEMS: A SURVEY
D400	SIGMOD	vol 22, 1993	A Survey on Usage of SQL
D401	SIGMOD	vol 22, 1993	An Update of the Temporal Database Bibliography
D402	SIGMOD	vol 22, 1993	Database Research at the Data-Intensive Systems Center
D403	The DATA BASE for Advances in Information Systems	vol 30, 1999	A Multi-Disciplinary Framework for the Management of Interorganizational Systems
D404	The DATA BASE for Advances in Information Systems	vol 30, 1999	Resource View Theory Analysis of SAP as a Source of Competitive Advantage for Firms
D405	The DATA BASE for Advances in Information Systems	vol 30, 1999	An Industry Analysis of Developer Beliefs About Object-Oriented Systems Development
D406	The DATA BASE for Advances in Information Systems	vol 30, 1999	IS "Maintainability":Should It Reduce the Maintenance Effort?

D407	The DATA BASE for Advances in Information Systems	vol 30, 1999	Application of Intelligent Agent Technology for Managerial Data Analysis and Mining
D408	The DATA BASE for Advances in Information Systems	vol 30, 1999	DataBase Special Issue- Information Systems: Current Issues and Future Changes
D409	The DATA BASE for Advances in Information Systems	vol 30, 1999	Co-Opetition and knowledge transfer
D410	The DATA BASE for Advances in Information Systems	vol 30, 1999	Knowledge capability and maturity in software management
D411	The DATA BASE for Advances in Information Systems	vol 30, 1999	Collaboration and Collaborative Information Technologies: A Review of the Evidence
D412	The DATA BASE for Advances in Information Systems	vol 30, 1999	Situated Assessment of Problems in Software Development
D413	The DATA BASE for Advances in Information Systems	vol 30, 1999	A Social Action Model of Situated Information Systems Design
D414	The DATA BASE for Advances in Information Systems	vol 30, 1999	Groupware Comes to the Internet:Charting a New World
D415	The DATA BASE for Advances in Information Systems	vol 30, 1999	Measuring Disagreement in Groups Facing Limited-Choice Problems
D416	The DATA BASE for Advances in Information Systems	vol 30, 1999	A Test of Task-TechnologyFit Theoryfor Group Support Systems
D417	The DATA BASE for Advances in Information Systems	vol 30, 1999	The Relation of Agenda Creation and Use to Group Support System Experience
D418	The DATA BASE for Advances in Information Systems	vol 30, 1999	The Facilitators Perspective on Meetings and Implications for Group Support Systems Design
D419	The DATA BASE for Advances in Information Systems	vol 30, 1999	Establishing a Foundation for Collaborative Scenario Elicitation
D420	The DATA BASE for Advances in Information Systems	vol 30, 1999	Exploring the Boundaries of Successful GSS Application: Supporting Inter-Organizational Policy Networks
D421	The DATA BASE for Advances in Information Systems	vol 30, 1999	Appropriations and Patterns In the Use of Group Support Systems
D422	The DATA BASE for Advances in Information Systems	vol 29, 1998	Introduction to the Special Issue…

D423	The DATA BASE for Advances in Information Systems	vol 29, 1998	The Effect of Data Model,System and Task Characteristics on User Query Performance - An Empirical Study
D424	The DATA BASE for Advances in Information Systems	vol 29, 1998	An Investigation of the Roles of Individual Differences and User Interface on Database Usability
D425	The DATA BASE for Advances in Information Systems	vol 29, 1998	Computer-Assisted Evaluation of Interface Designs
D426	The DATA BASE for Advances in Information Systems	vol 29, 1998	A Comparative Investigation of Ethical Decision Making: Information Systems Professionals versus Students
D427	The DATA BASE for Advances in Information Systems	vol 29, 1998	The Impact of Developer Responsiveness on Perceptions of Usefulness and Ease of Use: An Extension of the Technology Acceptance Model
D428	The DATA BASE for Advances in Information Systems	vol 29, 1998	LPL++: Object Logic Programming Language with Built-in Inheritance Through Unification
D429	The DATA BASE for Advances in Information Systems	vol 29, 1998	Revisiting the Perennial Question: Are IS People Different?
D430	The DATA BASE for Advances in Information Systems	vol 29, 1998	An MIS Course Integrating Information Technology and Organizational Issues
D431	The DATA BASE for Advances in Information Systems	vol 29, 1998	A Collaborative Fuzzy Expert System for the Web
D432	The DATA BASE for Advances in Information Systems	vol 29, 1998	An Empirical Validation of a Contingency Model for Information Require-ments Determination
D433	The DATA BASE for Advances in Information Systems	vol 29, 1998	Incremental View Maintenance in Object-Oriented Databases
D434	The DATA BASE for Advances in Information Systems	vol 29, 1998	SimDS: A Simulation Environment for the Design of Distributed Database Systems
D435	The DATA BASE for Advances in Information Systems	vol 29, 1998	Data Management Issues for Large Scale, Distributed Workflow Systems on the Internet
D436	The DATA BASE for Advances in Information Systems	vol 29, 1998	The DataIndex: A Structure for Smaller, Faster Data Warehouses
D437	The DATA BASE for Advances in Information Systems	vol 29, 1998	APSARA: A Tool to Auto-mate System Design via Intelligent Pattern Retrieval and Synthesis
D438	The DATA BASE for Advances in Information Systems	vol 29, 1998	ReconcilingTop-Down and Bottom-Up Design Approaches in RMM

D439	The DATA BASE for Advances in Information Systems	vol 29, 1998	A Comparative Case Study of Three Database Application Development Environments
D440	The DATA BASE for Advances in Information Systems	vol 28, 1997	Introduction to the Special Issue...Computers and Playfulness: Humorous,Cognitive, and Social Playfulness in Real and Virtual Workplaces
D441	The DATA BASE for Advances in Information Systems	vol 28, 1997	Microcomputer Playfulness: Stable or Dynamic Trait?
D442	The DATA BASE for Advances in Information Systems	vol 28, 1997	Individual Characteristics Associated with World Wide Web Use: An Empirical Study of Playfulness and Motivation
D443	The DATA BASE for Advances in Information Systems	vol 28, 1997	Audience Engagement in Multimedia Presentations
D444	The DATA BASE for Advances in Information Systems	vol 28, 1997	Personal Computer Adventure Games: Their Structure, Principles, and Applicability for Training
D445	The DATA BASE for Advances in Information Systems	vol 28, 1997	The Role of Work, Play, and Fun in Microcomputer Software Training
D446	The DATA BASE for Advances in Information Systems	vol 28, 1997	Examining the Relationship between Computer Cartoons and Factors in Information Systems Use, Success, and Failure." Visual Evidence of Met and Unmet Expectations
D447	The DATA BASE for Advances in Information Systems	vol 28, 1997	How Good is that Data in the Warehouse?
D448	The DATA BASE for Advances in Information Systems	vol 28, 1997	Is North American IS Research Different from European IS Research?
D449	The DATA BASE for Advances in Information Systems	vol 28, 1997	Development of a Measure to Assess the Quality of User-Developed Applications
D450	The DATA BASE for Advances in Information Systems	vol 28, 1997	A Total Quality Management-Based Systems Development Process
D451	The DATA BASE for Advances in Information Systems	vol 28, 1997	Introducing Client/Server Technologies in Information Systems Curricula
D452	The DATA BASE for Advances in Information Systems	vol 28, 1997	How Can CAS Help? A Look at the Feasibility of Supporting Structured Analysis with CASE
D453	The DATA BASE for Advances in Information Systems	vol 28, 1997	An Application of Rule-Based and Case-Based Reasoning within a Single Legal Knowledge-Based System
D454	The DATA BASE for Advances in Information Systems	vol 27, 1996	A Language-Oriented Data Modeling Approach

D455	The DATA BASE for Advances in Information Systems	vol 27, 1996	Factors Influencing Electronic Data Interchange Success
D456	The DATA BASE for Advances in Information Systems	vol 27, 1996	An Empirical Comparison of a Hypertext-Based Systems Analysis Case with Conventional Cases and Role Playing
D457	The DATA BASE for Advances in Information Systems	vol 27, 1996	An Assessment of Database Research Interest in MIS
D458	The DATA BASE for Advances in Information Systems	vol 27, 1996	Teaching Teamwork: Exploringthe Use of Cooperative Learning Teams in Information Systems Education
D459	The DATA BASE for Advances in Information Systems	vol 27, 1996	An Assessment of Structure and Causation of IS Usage
D460	The DATA BASE for Advances in Information Systems	vol 27, 1996	A Process Oriented Framework for Assessing the Business Value of Information Technology
D461	The DATA BASE for Advances in Information Systems	vol 27, 1996	From PRIISM. . .China's Movement Toward a National Information Infrastructure
D462	The DATA BASE for Advances in Information Systems	vol 27, 1996	The Use of Meta-Analysis in MIS Research: Promises and Problems
D463	The DATA BASE for Advances in Information Systems	vol 27, 1996	Emerging Issues in Interpretive Organizational Learning
D464	The DATA BASE for Advances in Information Systems	vol 27, 1996	User Perceptions of Evaluation Criteria for Three System Types
D465	The DATA BASE for Advances in Information Systems	vol 27, 1996	Practical Aspects of Teaching an Applied Database Course
D466	The DATA BASE for Advances in Information Systems	vol 27, 1996	Introduction to the Special issue...Forecasting the Next 50 Years in Information Technology
D467	The DATA BASE for Advances in Information Systems	vol 27, 1996	Information Futures: Producer and Consumer Views
D468	The DATA BASE for Advances in Information Systems	vol 27, 1996	IT: The Next 1100102 Years
D469	The DATA BASE for Advances in Information Systems	vol 27, 1996	Service: the Future of Information Technology
D470	The DATA BASE for Advances in Information Systems	vol 27, 1996	The Futures of IT Management

D471	The DATA BASE for Advances in Information Systems	vol 27, 1996	The Second Information Revolution
D472	The DATA BASE for Advances in Information Systems	vol 27, 1996	Networking: The Future of Information Systems
D473	The DATA BASE for Advances in Information Systems	vol 27, 1996	Artificial Intelligence and Gotterdamerung The Evolutionary Paradigm of the Future
D474	PODS90	1990	Representability of Design Objects by Ancestor-Controlled Hierarchical Specifications (Extended Abstract)
D475	PODS90	1990	Deriving Constraints Among Argument Sizes in Logic Programs
D476	PODS90	1990	Modular Stratification and Magic Sets for DATALOG Programs with Negation
D477	PODS90	1990	Three-Valued Formalization of Logic Programming: Is It Needed?
D478	PODS90	1990	On the Expressive Power of Datalog: Tools and a Case Study
D479	SIGMOD	vol 4, 1992	Query Optimization for Parallel Execution
D480	SIGMOD	vol 4, 1992	A Performance Analysis of Alternative Multi-Attribute Declustering Strategies
D481	SIGMOD	vol 4, 1992	Rule Condition Testing and Action Execution in Arielt
D482	SIGMOD	vol 4, 1992	Event Specification in an Active Object-Oriented Database
D483	PODS98	1998	Latent Semantic Indexing: A Probabilistic Analysis
D484	SIGMOD	vol 4, 1992	Performance Analysis of Coherency Control Policies through Lock Retention
D485	SIGMOD	vol 4, 1992	MLR: A Recovery Method for Multi-level Systems
D486	SIGMOD	vol 4, 1992	An Efficient Scheme for Providing High Availability
D487	SIGMOD	vol 4, 1997	Highly Concurrent Cache Consistency for Indices in Client-Server Database Systems
D488	SIGMOD	vol 4, 1997	Cubetree: Organization of and Bulk Incremental Updates on the Data Cube
D489	SIGMOD	vol 4, 1997	Maintenance of Data Cubes and Summary Tables in a Warehouse
D490	PODS	1998	Tight bounds for a-dimensional indexing schemes
D491	SIGMOD	vol 4, 1997	A Framework for Implementing Hypothetical Queries
D492	SIGMOD	vol 4, 1997	Partitioned Garbage Collection of a Large Object Store
D493	SIGMOD	vol 5, 1993	THE SEQUOIA 2000 STORAGE BENCHMARK
D494	SIGMOD	vol 5, 1993	Database System Issues in Nomadic Computing
D495	SIGMOD	vol 4, 1997	SEMCOG: An Object-based Image Retrieval System and Its Visual Query Interface
D496	SIGMOD	vol 5, 1993	Methods and Rules
D497	SIGMOD	vol 5, 1993	Using Shared Virtual Memory for Parallel Join Processing

D498	SIGMOD	vol 5, 1993	The Design and Implementation of CoBase
D499	ER	1998	Structure-Based Queries over the World Wide Web
D500	ER	1998	On the Consistency of Int-cardinality Constraints

Appendix C

The Testing Results

PDF No.	Title	Author	Abstract	Keyword	Subject	Converting Problems
D001	1.00	3/4	1.00	1	1/3	0
D002	1.00	1/2	0.50	1	1/4	0
D003	1.00	1/1	1.00	1	2/3	0
D004	1.00	1/2	1.00	1	1/2	0
D005	1.00	1/3	1.00	1	1/4	0
D006	1.00	1/2	1.00	1	1/3	0
D007	1.00	1/2	1.00	1	1/4	0
D008	1.00	1/1	1.00	N/A	1/3	0
D009	1.00	2/2	1.00	N/A	1/5	0
D010	1.00	1/1	1.00	1	1/3	0
D011	1.00	0/3	1.00	1	1/2	0
D012	1.00	3/4	0.50	1	2/5	0
D013	1.00	3/4	1.00	1	1/4	0
D014	1.00	3/3	1.00	1	1/3	0
D015	1.00	1/2	1.00	N/A	1	0
D016	1.00	2/2	1.00	1	1	0
D017	1.00	4/4	1.00	1	1/3	0
D018	1.00	3/4	1.00	1	1/3	0
D019	1.00	2/2	0.50	1	1/4	0
D020	1.00	0/2	1.00	1	2/5	0
D021	1.00	1/2	0.50	1	1/3	0
D022	1.00	4/4	1.00	1	1/2	0
D023	1.00	4/5	0.00	1	1/3	0

D024	1.00	3/3	1.00	1	1/3	0
D025	1.00	3/3	1.00	1	1/2	0
D026	1.00	2/3	1.00	1	1/3	0
D027	1.00	2/3	1.00	1	2/5	0
D028	1.00	4/4	0.00	1	3/4	0
D029	1.00	4/5	1.00	1	2/3	0
D030	1.00	2/3	1.00	1	1/2	0
D031	1.00	3/3	1.00	1	1/2	0
D032	1.00	3/3	0.50	1	2/3	0
D033	1.00	1/2	1.00	1	1	0
D034	1.00	3/3	0.00	1	1/3	0
D035	1.00	3/3	0.50	1	2/3	0
D036	1.00	1/2	1.00	1	1/4	0
D037	1.00	1/1	1.00	1	1/3	0
D038	1.00	3/3	0.00	1	1	0
D039	1.00	3/4	1.00	1	1	0
D040	1.00	2/3	0.00	0	1/4	0
D041	1.00	2/2	0.00	1	3/4	0
D042	1.00	3/3	0.50	0	3/4	0
D043	1.00	1/1	1.00	1	4/5	0
D044	1.00	3/3	0.50	1	1/2	0
D045	1.00	1/2	1.00	1	4/5	0
D046	1.00	2/3	1.00	1	1/3	0
D047	1.00	1/1	1.00	1	1/2	0
D048	1.00	2/2	1.00	1	1/4	0
D049	1.00	2/2	1.00	1	1/2	0
D050	1.00	1/2	1.00	1	1/4	0
D051	1.00	3/3	1.00	N/A	1/4	0

D052	1.00	3/3	1.00	1	2/3	0
D053	1.00	4/4	1.00	N/A	1/3	0
D054	1.00	4/4	1.00	1	1/4	0
D055	1.00	1/1	1.00	1	0	0
D056	1.00	3/3	1.00	1	2/3	0
D057	1.00	2/2	1.00	1	1/6	0
D058	1.00	1/1	1.00	1	1/2	0
D059	1.00	4/4	1.00	1	1/4	0
D060	0.00	2/2	1.00	0	1/3	0
D061	1.00	2/2	1.00	1	1/5	0
D062	1.00	1/2	1.00	1	1	0
D063	1.00	3/4	1.00	1	2/5	0
D064	1.00	1/2	1.00	1	1	0
D065	1.00	3/3	1.00	1	3/4	0
D066	1.00	4/4	1.00	1	1/2	0
D067	1.00	4/5	1.00	1	1/4	0
D068	1.00	2/2	1.00	1	1/2	0
D069	1.00	7/7	1.00	1	1/3	0
D070	1.00	2/2	1.00	1	1	0
D071	1.00	2/2	1.00	1	1	0
D072	1.00	3/3	1.00	1	1/4	0
D073	1.00	2/2	1.00	1	1/4	0
D074	1.00	4/4	1.00	1	0	0
D075	1.00	3/3	1.00	1	1	0
D076	1.00	3/3	1.00	1	1/2	0
D077	1.00	3/4	1.00	1	1/3	0
D078	1.00	1/2	1.00	1	1	0
D079	1.00	3/4	1.00	1	1/3	0
D080	1.00	2/2	1.00	1	1/2	0

D081	1.00	4/5	1.00	1	1/3	0
D082	1.00	5/5	1.00	1	2/3	0
D083	1.00	1/2	1.00	1	2/3	0
D084	1.00	1/3	1.00	1	1/4	0
D085	1.00	3/3	1.00	1	2/5	0
D086	0.00	4/4	1.00	1	2/3	0
D087	1.00	4/4	1.00	1	1/4	0
D088	1.00	3/3	1.00	1	1/3	0
D089	0.00	2/2	1.00	1	1/3	0
D090	1.00	1/2	1.00	1	1	0
D091	0.00	3/4	1.00	1	1	0
D092	0.00	1/2	1.00	1	1/2	0
D093	N/A	N/A	N/A	N/A	N/A	N/A
D094	1.00	3/6	1.00	1	1/3	0
D095	1.00	1/2	1.00	1	1/3	0
D096	N/A	N/A	N/A	N/A	N/A	N/A
D097	0.00	0/3	1.00	1	1/4	0
D098	1.00	0/2	1.00	1	1/2	0
D099	1.00	4/5	1.00	1	1/3	0
D100	0.00	1/2	1.00	1	1	0
D101	0.00	2/3	1.00	1	1/2	0
D102	1.00	2/2	1.00	1	2/3	0
D103	0.00	3/3	1.00	1	1/3	0
D104	1.00	3/3	1.00	1	1/4	0
D105	1.00	2/3	1.00	1	1/3	0
D106	0.00	1/1	1.00	1	1/2	0
D107	0.00	0/2	0.50	N/A	2/3	0
D108	1.00	2/2	1.00	1	1/3	0
D109	1.00	3/4	1.00	1	1	0

D110	1.00	1/4	1.00	1	1	0
D111	1.00	5/7	1.00	1	1/4	0
D112	0.00	1/2	1.00	1	1/3	0
D113	1.00	1/1	1.00	1	1/5	0
D114	1.000/1		1.00	1	1/3	0
D115	1.00	1/1	1.00	1	1/2	0
D116	1.00	2/3	1.00	1	1/2	0
D117	1.00	1/2	1.00	1	1/2	0
D118	1.00	2/3	1.00	1	2/3	0
D119	1.00	3/3	1.00	1	1	0
D120	0.00	1/2	1.00	1	1	0
D121	1.00	3/4	1.00	1	2/5	0
D122	1.00	3/3	1.00	1	1	0
D123	1.00	3/4	1.00	1	1/4	0
D124	1.00	3/3	1.00	1	1/4	0
D125	1.00	3/4	1.00	1	2/5	0
D126	1.00	2/2	1.00	1	1/3	0
D127	1.00	2/2	1.00	1	1/3	0
D128	1.00	1/1	1.00	1	1/3	0
D129	1.00	1/3	1.00	1	0	0
D130	1.00	2/3	1.00	1	1/2	0
D131	0.00	2/3	1.00	1	1/2	0
D132	0.00	2/2	1.00	1	2/3	0
D133	0.00	3/3	1.00	1	2/5	0
D134	1.00	2/2	1.00	1	1/4	0
D135	0.00	3/3	1.00	1	1	0
D136	0.50	1/1	1.00	1	1/4	0
D137	1.00	3/3	1.00	1	1/2	0

D138	1.00	3/3	1.00	1	1/3	0
D139	0.00	0/1	1.00	1	2/5	0
D140	1.00	5/7	0.00	0	0	0
D141	0.50	N/A	1.00	N/A	N/A	0
D142	1.00	5/5	1.00	1	1/2	0
D143	1.00	1/1	1.00	N/A	1/3	0
D144	1.00	2/2	1.00	N/A	0/3	0
D145	0.50	1/1	1.00	N/A	1/3	0
D146	N/A	N/A	N/A	N/A	N/A	0
D147	1.00	1/1	0.00	N/A	N/A	0
D148	1.00	1/1	1.00	0	1/2	0
D149	1.00	1/2	0.00	N/A	1/2	0
D150	1.00	2/2	1.00	N/A	1	0
D151	1.00	0/4	1.00	N/A	1/2	0
D152	1.00	N/A	1.00	N/A	N/A	0
D153	1.00	1/2	0.50	N/A	N/A	0
D154	N/A	N/A	N/A	N/A	N/A	1
D155	1.00	1/1	1.00	N/A	1/3	0
D156	N/A	N/A	N/A	N/A	N/A	1
D157	1.00	2/2	1.00	N/A	2/3	0
D158	N/A	N/A	N/A	N/A	N/A	1
D159	N/A	N/A	N/A	N/A	N/A	1
D160	1.00	3/4	1.00	N/A	2/3	0
D161	N/A	N/A	N/A	N/A	N/A	1
D162	N/A	N/A	N/A	N/A	N/A	1
D163	N/A	N/A	N/A	N/A	N/A	1
D164	0.50	1/2	0.50	N/A	N/A	0
D165	1.00	1/1	1.00	N/A	1/2	0

D166	1.00	8/9	1.00	N/A	1	0
D167	1.00	2/2	1.00	N/A	2/3	0
D168	1.00	1/1	1.00	N/A	1/3	0
D169	1.00	2/2	1.00	N/A	1/6	0
D170	1.00	3/3	1.00	N/A	3/4	0
D171	1.00	2/3	0.00	N/A	0/3	0
D172	1.00	3/4	1.00	0	1/4	0
D173	0.50	1/1	1.00	N/A	1/3	0
D174	1.00	3/3	1.00	N/A	2/3	0
D175	1.00	1/1	1.00	1	2/3	0
D176	1.00	1/1	1.00	N/A	0	0
D177	1.00	2/3	1.00	N/A	0	0
D178	N/A	N/A	N/A	N/A	N/A	0
D179	0.50	0/2	1.00	0	N/A	0
D180	1.00	2/2	1.00	N/A	1/3	0
D181	1.00	3/3	0.50	N/A	N/A	0
D182	1.00	5/7	1.00	N/A	N/A	0
D183	0.50	0/0	1.00	N/A	N/A	0
D184	0.50	2/2	1.00	1	2/3	0
D185	1.00	1/1	0.50	N/A	1/2	0
D186	1.00	0/3	0.50	N/A	0	0
D187						0
D188	0.50	2/3	1.00	N/A	1/2	0
D189	1.00	1/4	1.00	N/A	1/4	0
D190	1.00	2/2	1.00	N/A	0	0
D191	1.00	5/7	1.00	N/A	1	0
D192	1.00	1/1	1.00	N/A	2/3	0
D193	1.00	3/3	1.00	N/A	1/2	0
D194	1.00	1/1	1.00	N/A	1/3	0
D195	N/A	N/A	N/A	N/A	N/A	0
D196	1.00	1/1	1.00	N/A	1/3	0
D197	1.00	1/1	1.00	N/A	N/A	0

D198	1.00	2/2	1.00	N/A	0	0
D199	1.00	2/2	1.00	N/A	0	0
D200	1.00	2/2	1.00	N/A	N/A	0
D201	1.00	1/1	1.00	N/A	0	0
D202	1.00	1/1	0.00	N/A	0	0
D203	N/A	N/A	N/A	N/A	N/A	1
D204	N/A	N/A	N/A	N/A	N/A	1
D205	1.00	4/4	1.00	N/A	1/3	0
D206	N/A	N/A	N/A	N/A	N/A	1
D207	N/A	N/A	N/A	N/A	N/A	0
D208	N/A	N/A	N/A	N/A	N/A	0
D209	N/A	N/A	N/A	N/A	N/A	1
D210	N/A	N/A	N/A	N/A	N/A	1
D211	0.50	1/1	0.50	N/A	1/3	0
D212	1.00	1/1	1.00	N/A	1	0
D213	0.50	1/1	1.00	N/A	1/3	0
D214	N/A	N/A	N/A	N/A	N/A	1
D215	1.00	2/2	1.00	N/A	1/2	0
D216	1.00	2/2	1.00	N/A	1	0
D217	0.00	0/1	0.00	N/A	1/2	0
D218	N/A	N/A	N/A	N/A	N/A	0
D219	1.00	2/2	1.00	N/A	1/3	0
D220	1.00	3/3	1.00	N/A	1/4	0
D221	1.00	1/2	1.00	N/A	1/2	0
D222	1.00	1/3	1.00	N/A	1	0
D223	1.00	4/5	1.00	N/A	3/7	0
D224	N/A	N/A	N/A	N/A	N/A	0
D225	N/A	N/A	N/A	N/A	N/A	0
D226	1.00	2/3	1.00	N/A	1/3	0
D227	1.00	1/1	1.00	N/A	1/4	0
D228	1.00	1/2	1.00	N/A	1/2	0
D229	1.00	6/6	1.00	N/A	2/5	0

D230	1.00	2/2	1.00	N/A	1/4	0
D231	1.00	2/2	1.00	N/A	1/2	0
D232	0.50	1/3	1.00	N/A	1/2	0
D233	1.00	1/1	1.00	N/A	1/3	0
D234	1.00	1/2	1.00	N/A	1/3	0
D235	1.00	6/6	1.00	N/A	1/2	0
D236	N/A	N/A	N/A	N/A	N/A	0
D237	1.00	1/1	0.00	N/A	1/4	0
D238	1.00	2/2	1.00	N/A	1/3	0
D239	N/A	N/A	N/A	N/A	N/A	1
D240	N/A	N/A	N/A	N/A	N/A	1
D241	N/A	N/A	N/A	N/A	N/A	1
D242	1.00	1/1	1.00	1	1/3	0
D243	1.00	1/1	1.00	N/A	1/3	0
D244	N/A	N/A	N/A	N/A	N/A	1
D245	N/A	N/A	N/A	N/A	N/A	0
D246	1.00	2/2	1.00	N/A	1/2	0
D247	1.00	11/11	1.00	0	1/4	0
D248	1.00	1/1	1.00	1	1/3	0
D249	N/A	N/A	N/A	N/A	N/A	1
D250	0.00	N/A	0.00	N/A	1/3	0
D251	0.50	4/4	1.00	N/A	1/3	0
D252	N/A	N/A	N/A	N/A	N/A	1
D253	1.00	2/4	1.00	N/A	1/3	0
D254	1.00	2/2	1.00	N/A	1/2	0
D255	1.00	3/5	1.00	N/A	1/2	0
D256	1.00	2/2	1.00	N/A	1/4	0
D257	1.00	0/2	1.00	N/A	1/4	0
D258	N/A	N/A	N/A	N/A	N/A	0

D259	1.00	1/2	1.00	N/A	1/3	0
D260	1.00	3/5	1.00	N/A	1/3	0
D261	0.50	1/2	1.00	0	N/A	0
D262	1.00	1/3	1.00	N/A	1/3	0
D263	0.50	3/3	1.00	N/A	1/2	0
D264	N/A	N/A	N/A	N/A	N/A	0
D265	0.50	2/2	0.00	N/A	1/4	0
D266	1.00	2/7	1.00	0	N/A	0
D267	1.00	2/2	1.00	N/A	N/A	0
D268	1.00	1/1	1.00	N/A	N/A	0
D269	1.00	3/5	1.00	N/A	1/3	0
D270	1.00	1/1	1.00	N/A	1/3	0
D271	1.00	0/2	1.00	N/A	1/4	0
D272	1.00	0/2	1.00	N/A	1/4	0
D273	1.00	1/2	1.00	N/A	2/5	0
D274	1.00	2/3	1.00	N/A	1/3	0
D275	1.00	0/3	1.00	N/A	1/4	0
D276	1.00	4/6	1.00	N/A	1/3	0
D277	N/A	N/A	N/A	N/A	N/A	1
D278	1.00	0/2	1.00	N/A	0	0
D279	1.00	2/2		N/A	1/3	0
D280	0.50	1/1	1.00	N/A	1/2	0
D281	1.00	0/1	1.00	N/A	1/2	0
D282	1.00	3/3	1.00	N/A	1/3	0
D283	1.00	3/3	1.00	N/A	1/3	0
D284	1.00	3/3	1.00	N/A	0	0
D285	1.00	0/2	1.00	N/A	0	0
D286	1.00	2/2	1.00	N/A	1/3	0
D287	1.00	2/2	1.00	N/A	1/3	0

D288	1.00	2/2	0.50	N/A	1/4	0
D289	N/A	N/A	N/A	N/A	N/A	0
D290	1.000	/1	1.00	N/A	1/2	0
D291	1.000	/1	1.00	N/A	3/5	0
D292	1.000	/1	0.50	N/A	1/4	0
D293	1.00	2/3	1.00	N/A	2/5	0
D294	1.00	1/1	1.00	N/A	1/4	0
D295	1.00	2/2	1.00	N/A	1/5	0
D296	1.00	3/3	1.00	N/A	2/5	0
D297	1.00	1/3	1.00	N/A	1/3	0
D298	1.000	/5	1.00	N/A	1/2	0
D299	0.50	1/1	1.00	N/A	2/3	0
D300	0.50	1/1	0.50	N/A	1/3	0
D301	1.00	1/1	1.00	1	1/2	0
D302	1.00	4/4	1.00	N/A	1/4	0
D303	0.50	3/3	1.00	N/A	1/3	0
D304	1.00	1/1	1.00	N/A	1/3	0
D305	1.00	2/2	1.00	N/A	2/3	0
D306	1.00	1/1	1.00	N/A	1/2	0
D307	1.00	1/2	1.00	N/A	0	0
D308	1.00	1/1	1.00	N/A	1/2	0
D309	1.00	2/2	1.00	0	1	0
D310	N/A	N/A	N/A	N/A	N/A	1
D311	1.00	1/1	1.00	1	1	0
D312	N/A	N/A	N/A	N/A	N/A	1
D313	N/A	N/A	N/A	N/A	N/A	1
D314	1.00	4/4	1.00	N/A	0	0
D315	1.00	1/2	1.00	1	0	0
D316	N/A	N/A	N/A	N/A	N/A	1
D317	1.00	2/2	1.00	N/A	3/4	0
D318	N/A	N/A	N/A	N/A	N/A	0

D319	1.00	1/1	1.00	N/A	2/3	0
D320	N/A	N/A	N/A	N/A	N/A	0
D321	1.00	1/1	1.00	N/A	1/5	0
D322	1.00	2/2	1.00	N/A	1/4	0
D323	1.00	0/3	1.00	N/A	0	0
D324	1.00	3/5	1.00	N/A	1/2	0
D325	1.00	4/4	0.50	N/A	1/3	0
D326	1.00	0/3	0.50	N/A	1/4	0
D327	1.00	0/1	1.00	N/A	0	0
D328	1.00	3/3	1.00	N/A	1/2	0
D329	1.00	3/3	1.00	N/A	0	0
D330	1.00	1/1	1.00	N/A	2/3	0
D331	1.00	0/4	1.00	N/A	1/3	0
D332	1.00	2/2	1.00	N/A	0	0
D333	1.00	1/1	1.00	N/A	N/A	0
D334	1.00	4/4	1.00	N/A	N/A	0
D335	1.00	7/7	1.00	N/A	1	0
D336	1.00	10/10	1.00	0	1/2	0
D337	1.00	5/5	1.00	N/A	1	0
D338	1.00	0/1	1.00	N/A	1/5	0
D339	1.00	2/2	1.00	N/A	1/2	0
D340	1.00	2/4	0.50	N/A	N/A	0
D341	1.00	1/1	1.00	N/A	1	0
D342	1.00	1/1	0.00	N/A	N/A	0
D343	N/A	N/A	N/A	N/A	N/A	0
D344	1.00	0/1	1.00	N/A	N/A	0
D345	1.00	0/1	1.00	N/A	1/2	0
D346	1.00	3/3	1.00	N/A	1/3	0
D347	1.00	1/1	1.00	0	1/2	0
D348	1.00	1/1	1.00	N/A	2/3	0

D349	1.00	3/3	1.00	N/A	0	0
D350	1.00	0/1	1.00	N/A	1/4	0
D351	N/A	N/A	N/A	N/A	N/A	0
D352	0.50	1/1	1.00	N/A	0	0
D353	1.00	1/1	1.00	N/A	1/3	0
D354	1.00	2/2	1.00	N/A	0	0
D355	N/A	N/A	N/A	N/A	N/A	0
D356	1.00	1/2	1.00	N/A	1/4	0
D357	1.00	1/1	1.00	N/A	1/4	0
D358	1.00	3/3	0.50	N/A	1/5	0
D359	1.00	2/2	1.00	N/A	1/2	0
D360	1.00	3/3	1.00	N/A	0	0
D361	0.50	2/3	1.00	N/A	1/3	0
D362	1.00	3/5	1.00	N/A	1/3	0
D363	0.50	1/4	0.50	N/A	0	0
D364	1.00	0/2	0.50	N/A	1/2	0
D365	N/A	N/A	N/A	N/A	N/A	0
D366	1.00	1/1	1.00	N/A	1	0
D367	1.00	1/1	1.00	N/A	0	0
D368	1.00	2/2	1.00	N/A	1	0
D369	0.50	1/3	1.00	N/A	0	0
D370	1.00	2/2	1.00	N/A	1	0
D371	N/A	N/A	N/A	N/A	N/A	0
D372	1.00	0/1	1.00	N/A	N/A	0
D373	1.00	1/2	1.00	N/A	N/A	0
D374	1.00	0/2	0.00	N/A	0	0
D375	1.00	0/1	1.00	N/A	0	0
D376	1.00	N/A	1.00	N/A	1	0

D377	1.00	1/1	1.00	N/A	1	0
D378	1.00	1/1	1.00	N/A	N/A	0
D379	N/A	N/A	N/A	N/A	N/A	0
D380	1.00	1/1	1.00	N/A	N/A	0
D381	1.00	1/3	1.00	N/A	0	0
D382	1.00	1/1	1.00	N/A	0	0
D383	1.00	3/3	1.00	N/A	1/2	0
D384	1.00	2/2	1.00	N/A	1	0
D385	1.00	0/2	1.00	N/A	1	0
D386	1.00	2/3	1.00	N/A	N/A	0
D387	1.00	1/1	1.00	N/A	0	0
D388	1.00	1/1	1.00	N/A	1	0
D389	1.00	N/A	1.00	N/A	1	0
D390	1.00	0/1	1.00	0	1	0
D391	1.00	2/2	1.00	N/A	N/A	0
D392	1.00	1/5	1.00	N/A	1	0
D393	1.00	0/3	1.00	1	3/5	0
D394	1.00	1/1	1.00	1	2/3	0
D395	1.00	2/4	1.00	N/A	1/4	0
D396	1.00	1/1	1.00	N/A	1	0
D397	1.00	1/1	1.00	N/A	1/4	0
D398	1.00	2/2	1.00	1	N/A	0
D399	1.00	1/2	1.00	N/A	2/3	0
D400	1.00	1/3	1.00	N/A	1/2	0
D401	1.00	1/1	1.00	N/A	1	0
D402	1.00	3/4	1.00	1	1	0
D403	1.00	2/2	1.00	1	1/4	0
D404	1.00	1/1	1.00	1	0	0

D405	1.00	3/3	1.00	1	3/4	0
D406	1.00	1/1	1.00	1	1/3	0
D407	1.00	2/2	1.00	1	1/4	0
D408	0.50	2/2	0.00	1	N/A	0
D409	1.00	3/3	1.00	1	0	0
D410	1.00	2/2	1.00	1	0	0
D411	1.00	1/1	1.00	1	1/2	0
D412	1.00	2/3	1.00	1	1/2	0
D413	1.00	1/1	1.00	1	1/2	0
D414	1.00	2/3	1.00	1	2/3	0
D415	1.00	2/2	1.00	1	1	0
D416	1.00	4/4	1.00	1	1/2	0
D417	1.00	2/2	1.00	1	1/2	0
D418	1.00	1/1	1.00	1	1/2	0
D419	1.00	1/1	1.00	1	1/2	0
D420	1.00	2/2	1.00	1	1/2	0
D421	1.00	1/1	1.00	1	1/2	0
D422	1.00	2/2	0.00	N/A	N/A	0
D423	1.00	1/3	1.00	1	1	0
D424	1.00	3/3	1.00	1	1/4	0
D425	1.00	1/1	1.00	1	1/3	0
D426	1.00	2/2	1.00	1	1/2	0
D427	1.00	2/2	1.00	1	1/4	0
D428	1.00	1/1	1.00	1	3/4	0
D429	1.00	2/2	1.00	1	0	0
D430	1.00	0/1	1.00	1	0	0
D431	1.00	1/1	1.00	1	0	0
D432	1.00	3/3	1.00	1	1/2	0
D433	1.00	2/2	1.00	1	0	0

D434	1.00	3/3	1.00	1	1/3	0
D435	1.00	2/2	1.00	1	1/4	0
D436	1.00	1/2	1.00	N/A	0	0
D437	1.00	1/1	0.50	N/A	1/2	0
D438	1.00	2/3	1.00	1	0	0
D439	1.00	2/2	1.00	1	0	0
D440	0.50	2/2	0.00	N/A	N/A	0
D441	1.00	3/4	0.50	N/A	0	0
D442	1.00	1/2	1.00	1	1/4	0
D443	1.00	2/2	1.00	1	1/4	0
D444	0.50	1/2	1.00	1	1/2	0
D445	0.50	2/2	1.00	1	0	0
D446	1.00	1/1	1.00	1	0	0
D447	1.00	1/1	1.00	1	N/A	0
D448	1.00	2/2	1.00	1	N/A	0
D449	1.00	3/4	1.00	1	0	0
D450	1.00	3/3	1.00	1	N/A	0
D451	1.00	2/2	1.00	1	N/A	0
D452	1.00	1/1	1.00	1	N/A	0
D453	1.00	1/2	1.00	1	1/5	0
D454	1.00	2/2	1.00	1	0	0
D455	1.00	1/2	1.00	1	0	0
D456	1.00	3/4	1.00	1	1/7	0
D457	1.00	1/1	0.00	1	1/2	0
D458	1.00	1/1	1.00	1	0	0
D459	1.00	2/2	1.00	1	0	0
D460	0.50	3/3	1.00	1	0	0
D461	0.50	1/1	1.00	1	N/A	0
D462	1.00	1/1	1.00	1	N/A	0

D463	1.00	3/3	1.00	1	1/3	0
D464	1.00	2/2	1.00	1	1/3	0
D465	1.00	2/2	1.00	1	1/2	0
D466	0.50	1/1	0.00	N/A	N/A	0
D467	1.00	1/1	1.00	N/A	0	0
D468	1.00	2/2	1.00	N/A	0	0
D469	1.00	3/3	1.00	N/A	0	0
D470	1.00	1/1	1.00	1	0	0
D471	1.00	1/1	1.00	N/A	0	0
D472	1.00	2/2	1.00	1	1/3	0
D473	1.00	1/1	1.00	1	1	0
D474	1.00	2/2	1.00	N/A	2/3	0
D475	1.00	1/2	0.50	N/A	1/2	0
D476	1.00	1/1	1.00	N/A	0	0
D477	1.00	2/2	1.00	N/A	1/2	0
D478	1.00	1/2	1.00	N/A	2/7	0
D479	1.00	1/3	1.00	N/A	2/5	0
D480	1.00	2/2	0.00	N/A	N/A	0
D481	1.00	2/2	1.00	N/A	N/A	0
D482	1.00	2/3	1.00	0	N/A	0
D483	1.00	4/4	1.00	N/A	0	0
D484	1.00	2/2	1.00	N/A	N/A	0
D485	1.00	1/3	1.00	N/A	N/A	0
D486	1.00	2/2	1.00	N/A	N/A	0
D487	1.00	2/2	1.00	N/A	2/5	0
D488	1.00	2/2	1.00	N/A	0	0
D489	1.00	2/2	1.00	N/A	0	0

D490	1.00	2/2	2.00	N/A	0	0
D491	1.00	2/2	1.00	N/A	0	0
D492	1.00	1/1	1.00	N/A	1/2	0
D493	1.00	3/4	1.00	N/A	1/3	0
D494	1.00	2/2	1.00	N/A	0	0
D495	1.00	1/1	1.00	N/A	2/3	0
D496	0.50	2/2	1.00	N/A	0	0
D497	1.00	3/3	1.00	N/A	1/5	0
D498	1.00	2/2	1.00	N/A	0	0
D499	0.50	3/3	1.00	N/A	0	0
D500	1.00	1/1	1.00	N/A	0	0
Average Accuracy(%)	92	80	92	93	41	95

References

[1] B. C. Desai, "The Semantic Header Indexing and Searching on the Internet, Department of Computer Science", Concordia University, Montreal, Canada, February 1995, http://www.cs.concordia.ca/ faculty/bcdesai/cindi-system-1.1.html

[2] M. Kobayashi, K. Takeda, "Information Retrieval on the Web", *ACM Computing Surveys*, Vol. 32, No. 2, pp144-173, June 2000

[3] B. C. Desai, S. Rajjan, "A System for Seamless Search of Distributed Information Sources", May 1994, http://www.cs.concordia.ca/~faculty/bcdesai

[4] W. A. Katz, *Introduction to Reference Work.* Vol. 1-2, New York: McGraw-Hill, 2001,

[5] N. Shayan, "CINDI: Concordia INdexing and DIscovery system", M.S. thesis, Department of Computer Science, Concordia University, Montreal, Canada, 1997.

[6] S. S. Haddad, "Automatic Semantic Header Generator", M.S. thesis, Department of Computer Science, Concordia University, July, 1998.

[7] G. Salton and M. J. Mcgill, *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., Hightstown, NJ. 1983.

[8] J. H. Hayes, A. Dekhtyar, J. Osborne, "Improving Requirements Tracing via Information Retrieval", *Proceedings of the 11th IEEE International Requirements Engineering Conference*, pp 1-10, 2003

[9] R. Baeza-Yates, B. Ribeiro-Neto, *Modern Information Retrieval*, Reading, Mass. Addison-Wesley Longman, 1999

[10] L. B. Wang, B. Fan, H. Yang, "The design and implementation of the Chinese information retrieval with the automatically indexing method", *IEEE Transactions on Applied Superconductivity*, pp 851-854, 2003

[11] D. B. Leake, R. Scherle, "Towards Context-Based Search Engine Selection", *International Conference on Intelligent User Interfaces,* ACM, pp 109-112, 2001

[12] A. Arasu, J. Cho, H. Garcia-Molina, A. Paepcke and S. Raghavan, "Searching the Web", *ACM Transactions on Internet Technology*, 1(1), 2001.

[13] S. Lawrence, and C. Giles, "Accessibility of information on the web", *Nature 400*, pp 107–109, 1999

[14] W3 Consortium, "Resource Description Framework", http://www.w3.org/RDF/

[15] K. Nagao, and K. Hasida, "Automatic text summarization based on the global document annotation", in Proceedings of the Conference on COLING-ACL, 1998.

[16] K. Nagao, S. Hosoya, Y. Kawakita, S. Ariga, Y. Shirai, and J. Yura, Semantic transcoding: Making the world wide web more understandable and reusable by external annotations, 1999.

[17] W. Cathro, "Matching discovery and recovery", in *Proceedings of the Seminar on Standards Australia*, www.nla.gov.au/staffpaper/cathro3.html, 1997

[18] C. Lagoze, "The Warwick framework: A container architecture for diverse sets of metadata". D-Lib Mag. www.dlib.org, 1996

[19] P. De Bra, G-J. Houben, and Y. Kornatzky, "Search in the World-Wide Web", http://www.win.tue.nl/help/doc/demo.ps

[20] J. Fletcher, "Jump station", http://www.stir.ac.uk/jsbin/js, 1993.

[21] M. Koster, "ALIWEB (Archie Like Indexing the WEB)", http://web.nexor.co.uk/aliweb/doc/aliweb.html

[22] Oliver A. McBryan, "World Wide Web Worm", http://www.cs.colorado.edu/home/mcbryan/WWWW.html

[23] Experimental Search Engine Meta-Index, http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Demo/metaindex.html

[24] R. Thau, SiteIndex Transducer, http://www.ai.mit.edu/tools/site-index.html

[25] Search WWW document full text, http://rbse.jsc.nasa.gov/eichmann/urlsearch.html

[26] WebCrawler, http://www.biotech.washington.edu/WebCrawler/WebQuery.html

[27] World Wide Web Catalog, http://cuiwww.unige.ch/cgi-bin/w3catalog

[28] B. C. Desai, "Cover page aka Semantic Header", http: // www.cs.concordia.ca/semantic-header.html, revised version, August 1994, http://www.cs.concordia.ca/ bcdesai/semantic-header.html

[29] Bipin C. Desai: "Supporting Discovery in Virtual Libraries", JASIS *48(3)*, pp 190-204, 1997

[30] Adobe Solutions Network, "PDF Reference", http://partners.adobe.com/asn/tech/pdf/specifications.jsp

[31] U. Pfeifer, T. Poersch and N. Fuhr, "Retrieval effectiveness of proper name search methods", *Information Processing & Management*, Vol.32, No.6, pp.667-679, 1996

[32] J. D. Burger, J. S. Aberdeen, D. D. Palmer, "Information Retrieval and Trainable Natural Language Processing", in *Proceedings of the fifth Test Retrieval Conference*, Gaithersburg, Maryland, November 20-22, 1996.

[33] Inspec, The Database for Physics, Electronics and Computing, http://www.iee.org/publish/inspec/

[34] ACM anthology, http://cindi2.concordia.ca/bcdesai/acm/

[35] Zhang, Z. "Porting the automatic semantic header generator to the web", M. S thesis, Department of Computer Science, Concordia University, July, 1998.