An E-Librarian Service

Supporting Explorative Learning by a Description Logics Based Semantic Retrieval Tool

Elektronisch veröffentlicht auf dem
Publikationsserver der Universität Potsdam:
http://opus.kobv.de/ubp/volltexte/2008/1745/
urn:nbn:de:kobv:517-opus-17452
[http://nbn-resolving.de/urn:nbn:de:kobv:517-opus-17452]

Hasso-Plattner-Institut für Software- und Systemtechnik GmbH
an der Universität Potsdam

An E-Librarian Service

Supporting Explorative Learning by a Description Logics Based Semantic Retrieval Tool

Dissertation
zur Erlangung des akademischen Grades
"doctor rerum naturalium"
(Dr. rer. nat.)
am Fachgebiet Internet Technologien und Systeme

eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät
der Universität Potsdam

von
Serge Linckels

Potsdam, September 2007

Gutachter:
Prof. Dr. Christoph Meinel, Hasso-Plattner-Institut (HPI)
Prof. Dr. Thomas Engel, University of Luxembourg
Prof. Dr.-Ing. Djamshid Tavangarian, University of Rostock

Prüfungskommission:
Prof. Dr. Felix Naumann, Hasso-Plattner-Institut (HPI), Vorsitzender
Prof. Jean-Claude Asselborn, University of Luxembourg
Prof. Dr. Dr. hc. Klaus Denecke, University of Potsdam
Prof. Dr. Andreas Polze, Hasso-Plattner-Institut (HPI)
Prof. Dr. Torsten Schaub, University of Potsdam
Prof. Dr. Andreas Schwill, University of Potsdam
Prof. Dr. Mathias Weske, Hasso-Plattner-Institut (HPI)

Datum der Disputation: 31.03.2008, Bewertung: *summa cum laude.*

Zusammenfassung

Obwohl sich die Verfügbarkeit von pädagogischen Inhalten in elektronischer Form stetig erhöht, ist deren Nutzen in einem schulischen Umfeld recht gering. Die Hauptursache dessen ist, dass es zu viele unzuverlässige, redundante und nicht relevante Informationen gibt. Das Finden von passenden Lernobjekten ist eine schwierige Aufgabe, die vom benutzerbasierten Filtern der passenden Informationen abhängig ist. Damit Wissensbanken wie das online Tele-TASK Archiv zu nützlichen, pädagogischen Ressourcen werden, müssen Lernobjekte korrekt, zuverlässig und in maschinenverständlicher Form identifiziert werden, sowie effiziente Suchwerkzeuge entwickelt werden.

Unser Ziel ist es, einen E-Bibliothekar-Dienst zu schaffen, der multimediale Ressourcen in einer Wissensbank auf effizientere Art und Weise findet als mittels Navigieren durch ein Inhaltsverzeichnis oder mithilfe einer einfachen Stichwortsuche. Unsere Prämisse ist, dass passendere Ergebnisse gefunden werden könnten, wenn die semantische Suchmaschine den Sinn der Benutzeranfrage verstehen würde. In diesem Fall wären die gelieferten Antworten logische Konsequenzen einer Inferenz und nicht die einer Schlüsselwortsuche.

Tests haben gezeigt, dass unser E-Bibliothekar-Dienst unter allen Dokumenten in einer gegebenen Wissensbank diejenigen findet, die semantisch am besten zur Anfrage des Benutzers passen. Dabei gilt, dass der Benutzer eine vollständige und präzise Antwort erwartet, die keine oder nur wenige Zusatzinformationen enthält. Außerdem ist unser System in der Lage, dem Benutzer die Qualität und Pertinenz der gelieferten Antworten zu quantifizieren und zu veranschaulichen. Schlussendlich liefert unser E-Bibliothekar-Dienst dem Benutzer immer eine Antwort, selbst wenn das System feststellt, dass es keine vollständige Antwort auf die Frage gibt.

Unser E-Bibliothekar-Dienst ermöglicht es dem Benutzer, seine Fragen in einer sehr einfachen und menschlichen Art und Weise auszudrücken, nämlich in natürlicher Sprache. Linguistische Informationen und ein gegebener Kontext in Form einer Ontologie werden für die semantische Übersetzung der Benutzereingabe in eine logische Form benutzt.

Unser E-Bibliothekar-Dienst wurde prototypisch in drei unterschiedliche pädagogische Werkzeuge umgesetzt. In zwei Experimenten wurde in einem pädagogischen Umfeld die Angemessenheit und die Zuverlässigkeit dieser Werkzeuge als Komplement zum klassischen Unterricht geprüft. Die Hauptergebnisse sind folgende: Erstens wurde festgestellt, dass Schüler generell akzeptieren, ganze Fragen einzugeben - anstelle von Stichwörtern - wenn dies ihnen hilft, bessere Suchresultate zu erhalten. Zweitens, das wichtigste Resultat aus den Experimenten ist die Erkenntnis, dass Schuleresultate verbessert werden können, wenn Schüler unseren E-Bibliothekar-Dienst verwenden. Wir haben eine generelle Verbesserung von 5% der Schulresultate gemessen. 50% der Schüler haben ihre Schulnoten verbessert, 41% von ihnen sogar maßgeblich. Einer der Hauptgründe für diese positiven Resultate ist, dass die Schüler motivierter waren und folglich bereit waren, mehr Einsatz und Fleiß in das Lernen und in das Erwerben von neuem Wissen zu investieren.

Overview

Although educational content in electronic form is increasing dramatically, its usage in an educational environment is poor, mainly due to the fact that there is too much of (unreliable) redundant, and not relevant information. Finding appropriate answers is a rather difficult task being reliant on the user filtering of the pertinent information from the *noise*. Turning knowledge bases like the online tele-TASK archive into useful educational resources requires identifying correct, reliable, and "machine-understandable" information, as well as developing simple but efficient search tools with the ability to reason over this information.

Our vision is to create an *E-Librarian Service*, which is able to retrieve multimedia resources from a knowledge base in a more efficient way than by browsing through an index, or by using a simple keyword search. In our E-Librarian Service, the user can enter his question in a very simple and human way; in natural language (NL). Our premise is that more pertinent results would be retrieved if the *search engine* understood the sense of the user's query. The returned results are then logical consequences of an inference rather than of keyword matchings. Our E-Librarian Service does not return the answer to the user's question, but it retrieves the most pertinent document(s), in which the user finds the answer to his/her question.

Among all the documents that have some common information with the user query, our E-Librarian Service identifies the most pertinent match(es), keeping in mind that the user expects an exhaustive answer while preferring a concise answer with only little or no information overhead. Also, our E-Librarian Service always proposes a solution to the user, even if the system concludes that there is no exhaustive answer.

Our E-Librarian Service was implemented prototypically in three different educational tools. A first prototype is CHESt (*Computer History Expert System*); it has a knowledge base with 300 multimedia *clips* that cover the main events in computer history. A second prototype is MatES (*Mathematics Expert System*); it has a knowledge base with 115 clips that cover the topic of fractions in mathematics for secondary school w.r.t. the official school programme. All clips were recorded mainly by pupils. The third and most advanced prototype is the "Lecture Butler's E-Librarain Service"; it has a Web service interface to respect a service oriented architecture (SOA), and was developed in the context of the Web-University project at the Hasso-Plattner-Institute (HPI).

Two major experiments in an educational environment — at the Lycée Technique Esch/Alzette in Luxembourg — were made to test the pertinence and reliability of our E-Librarian Service as a complement to traditional courses. The first experiment (in 2005) was made with CHESt in different classes, and covered a single lesson. The second experiment (in 2006) covered a period of 6 weeks of intensive use of MatES in one class. There was no classical mathematics lesson where the teacher gave explanations, but the students had to learn in an autonomous and exploratory way. They had to ask questions to the E-Librarian Service just the way they would if there was a human teacher.

Acknowledgements

This thesis could not have been written without the support of lots of people. First of all, I would like to thank my doctor-father Prof. Dr. Christoph Meinel — who gave me the opportunity to start as an external Ph.D. student and always had an open door for me — for all the interesting discussions and comments. A lot of ideas that I elaborated on this thesis originated from our discussions. Prof. Meinel taught me a lot of things beyond computer science and turned the years of my Ph.D. work into an amazing adventure.

I feel a deep sense of gratitude towards my wife Magali for providing technical help and advice, and for continuously encouraging me in moments of stress and resignation.

I also want to express my gratitude to Prof. Dr. Djamshid Tavangarian, Prof. Dr. Thomas Engel and Prof. Dr. Abraham Bernstein who have accepted to review this thesis, to Isabelle Wirth for correcting my English errors in this document, and to Carole Dording who spent uncountable hours on elaborating the knowledge base of MatES and on supervising the experiment in school.

This research project was supported by *Luxembourg International Advanced Studies in Information Technologies* (LIASIT). Therefore, my special thanks goes to Prof. Dr. Thomas Engel and the whole LIASIT "family" for their support.

I want to express my gratitude to the former and to today's board of directors of the *Lycée Technique Esch/Alzette* (LTE) for supporting this research project and for being part of it as "external partner". In particular, I am grateful for being granted the authorization to test some of the prototypes in a real school environment.

I would also like to thank Dr. Naouel Karam, Dr. Laurent Debrauwer and Dr. Harald Sack for all the interesting meetings, explanations, reviews and comments, and Monique Reichert for the preparation and evaluation of the CHESt experiment.

Besides all the personal acknowledgements, I would like to thank all the (former) staff of the chair "Internet Technologies- and Systems" at the Hasso-Plattner-Institute for their help, encouragement, and the numerous rewarding moments we spent together. All the people with whom I had the chance to interact in the course of this project have greatly enriched my life.

Finally, I want to express my gratitude towards my parents who have always believed in me and encourage me accordingly.

Table of Contents

Chapter 1

Introduction

Contents

Libraries were throughout our history always the carrier of knowledge, and the instrument for learning from the experiences and the endeavors of previous generations. Libraries are essential for a well informed citizenry and transparent governance. More recently, libraries are understood as extending beyond the physical walls of a building, by including material accessible by electronic means, and by providing the assistance of librarians in navigating and analyzing tremendous amounts of knowledge with a variety of digital tools. The ambition of this project is to conceive a virtual librarian — we call it an *E-Librarian Service* — that is able to find the most appropriated resource in its repository.

This chapter is organized as follows. Section 1.1 will give a short introduction to (digital) libraries and librarians, and their role in education. We will formulate in section 1.2 our vision to create an E-Librarian Service, and emphasize its major characteristics. Section 1.3 will summarize the contributions while section 1.4 will give an overview of the structure of the thesis.

1.1 From Ancient Libraries to Librarians in Education

The aim of this section is to emphasize the importance of libraries and librarians throughout history. We will start with a brief historical overview before describing the necessity of libraries in education.

1.1.1 History of Libraries and Librarians

The collection of written knowledge in some sort of repository is a practice as old as civilization itself. About 30,000 clay tablets found in ancient Mesopotamia date back more than 5,000 years. Archeologists have uncovered papyrus scrolls from 1300 – 1200 BC in the ancient Egyptian cities of Amarna and Thebes, and thousands of clay tablets in the palace of King Sennacherib, Assyrian ruler from 704 – 681 BC at Nineveh, his capital city. The name for the repository eventually became “the library”, which means in Greek: “collection of books”. The Latin word for library is: “bibliotheca”.

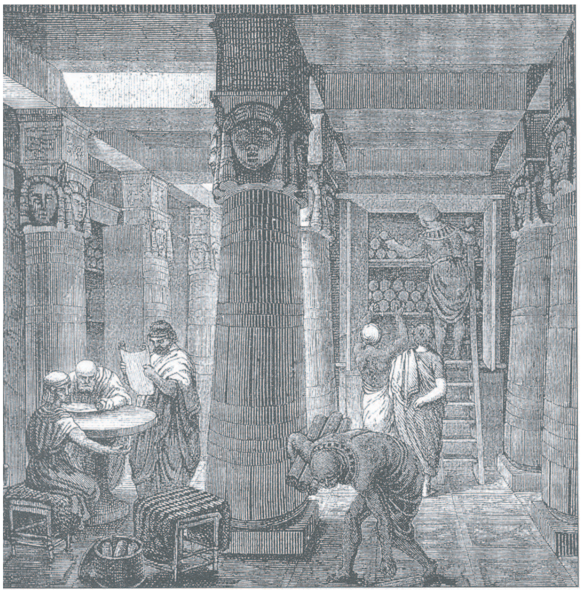


Figure 1.1: Artistic Rendering of “The Great Library of Alexandria” by O. Von Corven.

Early collections of papyrus — usually viewed as archives rather than libraries — may have surfaced from the Near East, but the ancient Greeks propelled the idea through their heightened interest in literacy and intellectual life. Public and private libraries flourished through a well-established process: authors wrote on a variety of subjects, scriptoria or copy shops produced the books, and book dealers sold them. Copying books was an exacting business and one in high demand, because a book’s “trustworthiness” translated into quality. An Athenian decree called for a repository of “trustworthy” copies. The early word for book was “codex” (plural *codices*), which was a Roman invention that replaced the *scroll*. A codex (Latin for *block of wood*) is a book in the format used for modern books, with separate pages normally bound together and given a cover.

Throughout the 1600s and 1700s, libraries surged in popularity. They grew as universities developed[1], and as national state-supported collections began to appear. A national library is a library specifically established by the government of a country to serve as the preeminent repository

[1] http://www.ib.hu-berlin.de/~pz/zahnpage/librdisc.htm

of information for that country. Three libraries form the national repository for Germany. The first, the German State Library in Berlin was founded in 1661. The second and third followed much later: the German Library in Leipzig was founded in 1912, and the German Library in Frankfurt was founded in 1946.

The 20^{th} century saw the continued development of the library through education and organization. In 1928 the first Ph.D. in Library Science was awarded at the University of Chicago. Huge changes were on the horizon for the profession as first microforms in the 1930s and 1940s, and then electronic databases in the 1950s and 1960s appeared. The library profession was becoming increasingly technical.

The further growth in electronic media available to the general public and its supposed ease of use have caused many to claim that librarianship as a profession will soon be obsolete. But the number of students attending library schools has increased over the last ten years and through changes in curricula library schools seem to be adapting to the new information landscape.

Throughout history, many people who later became well known in other capacities served as librarians. Here are some examples:

- Gottfried von Leibniz (1646 – 1716) was a German philosopher, mathematician, and intellectual giant of his time. Leibniz was appointed librarian at Hannover in 1676 and at Wolfenbuttel in 1691.

- Giacomo Casanova (1725 – 1798) was not only a great lover. At the climax of his career in 1785, the famous womanizer began to work for 13 years as a librarian for the Count von Waldstein in the chateau of Dux in Bohemia.

- Mao Tse-Tung (1893 – 1976) worked as an assistant to the chief librarian of the University of Peking. Overlooked for advancement, he decided to get ahead in another field and eventually became chairman of the Chinese Communist Party.

- FBI Head J. Edgar Hoover (1895 – 1972) was a Library of Congress messenger and cataloger in his first job.

Today, we witness a tremendous increase in the availability of information throughout knowledge repositories in digital form. At the Hasso-Plattner-Institut (HPI) in Potsdam alone, 25 hours of university lecture videos about computer science are produced every week. Most of them are published at the online Tele-TASK archive[2]. The term *digital library* was first made popular by the NSF-DARPA-NASA Digital Libraries Initiative in 1994.

1.1.2 Libraries and Education

The unique task of libraries and information services is that they respond to the particular questions and needs of individuals. This complements the general transmission of knowledge by the media, and makes libraries and information services vital to a democratic and open Information Society. Libraries are essential for a well informed citizenry and transparent governance.

Libraries in education institutions have developed a wide range of services to meet the educational objectives of their parent institutions. School libraries clearly need to support the curriculum, but they also collect books and other materials to encourage reading and spirit of enquiry, as well as to meet the needs of the teachers and administrative staff.

[2] `http://www.tele-task.de/`

Libraries were throughout our history always the carrier of knowledge, and the instrument for learning from the experiences and the endeavors of previous generations. Libraries are needed more than ever in an age in which people and communities desperately need to consider alternative points of view and information, to challenge the spin doctors and the mass media, to take control of their own destinies, and make up their own minds.

Educational Institutions, such as universities, colleges, and schools all have libraries serving the educational objectives of their parent bodies. But the simple existence of a library is not sufficient for people to learn. Students must be guided in their learning process by qualified people, who show them how and what to learn [Mar03, FDD$^+$99, NPSR99, Blo01, Mor05]. This is mostly the task of teachers, but also of experts helping the learners to access knowledge repositories, and to find appropriate information. Thus, users need to access materials through libraries, which have skilled staff to search efficiently, and are able to identify authentic resources.

1.2 Towards an E-Librarian Service

Motivated by the importance of libraries and librarians presented above, we will describe in this section our vision of an E-Librarian Service.

1.2.1 The Librarian's Problem

Let us suppose that Paul wants to learn about the invention of the transistor. He goes to a library and asks the librarian: "I want to know who invented the transistor?" The librarian perfectly understood Paul's question and knows where to find the right book. (S)He also understood that Paul does not want all available books in the library that explain how a transistor works, or those which illustrate in detail the lives of its inventor(s). It is evident for the librarian that Paul only wants one pertinent document in which he will find the answer to his question. In conclusion, we can make the following statements:

For the client:

- Paul formulates his question in natural language,
- Paul has no knowledge about the internal organization of the books in the library,
- Paul does not know what he is looking for in particular; he gave no book title to the librarian.

For the librarian:

- (s)he is able to understand the client's question (the language and the sense),
- (s)he does not know the answer to the client's question,
- (s)he controls the internal organization of the library,
- from all the existing books in the library, (s)he finds the one(s) that best fit(s) the needs of the client.

It is obvious that the larger the library is, the more books will be potentially pertinent. If Paul wants to be sure that he will only get a very short list of relevant books, then he should go to a specialized library. There, the potential amount of documents is far smaller but the chance to find pertinent results is higher. Visiting specialized libraries also reduces the risk of ambiguity. If Paul asked in a general library for a book about golf, the librarian would have the choice between several possible interpretations: a sport, a German car, a geographical position. However, if Paul was in a library about sports, the context would be clear.

1.2.2 The Vision of an E-Librarian Service

Our vision is to create an *E-Librarian Service*, a computer based expert system that offers the same services as a real librarian. One should not confuse it with a software to manage a library, or with a search engine over a catalogue. Unlike classical search engines or question-answering systems, our E-Librarian Service does not deliver the answer to the user's question, but it is able to find and retrieve the most pertinent document. The user will find the answer to his/her question in that document. We summarize the characteristics of our E-Librarian Service as follows:

- it has a huge amount of stored knowledge in multimedia form,
- it controls the internal organization of its knowledge base,
- it "understands" the sense of the users' questions,
- it finds pertinent documents in its knowledge base with respect to a user's query,
- it is simply accessible without complicated software or hardware requirements,
- it is simple to use, i.e., interaction in a human way by means of verbal communication,
- it is able to visualize the pertinence of the delivered documents, i.e., ranking of the results according to their semantic relatedness to the query.

In this thesis, we will explore strategies, and novel and promising technologies from different research fields — e.g., Semantic Web, computational linguistics, multimedia information retrieval, ontologies, and Description Logics — to conceive, implement, and evaluate such an E-Librarian Service.

1.3 Objective and Contributions

This versatile project focuses on the design, elaboration, and testing of a novel approach of retrieval systems by bringing together the most appropriated knowledge and technologies from different research domains. The development of our E-Librarian Service — a reliable and efficient tool to quickly find pertinent resources in a multimedia repository — is fully in stream of the Semantic Web philosophy, and joins the efforts to standardize reusable learning objects, ontologies, and technologies.

The results of this research work are, firstly, a founded background theory that improves domain search engines to be able to retrieve semantically pertinent documents from a multimedia repository, based on the semantic interpretation of a complete question expressed in NL into a logical form, i.e., Description Logics. Secondly, we will provide empirical data that prove the feasibility, and the effectiveness of our E-Librarian Service. This data was collected by benchmark tests, but also by experiments made in an educational environment with different prototypes of our E-Librarian Service. A detailed list of key results is presented in chapter 13. Finally, we will show that students perceive our E-Librarian Service as an efficient and helpful tool that can be used as a complement to traditional courses, in class or at home.

Academy benefits The solution that we will present improves domain ontology search engines; fewer results, but more pertinent ones are returned. This project will also contribute to current ontology research by providing a rich and documented schemata for representing ontologies about computer history, mathematics and networking in computer science. The thesis contributes to the discussions, in how far a search engine would yield better results if the query was formulated in NL, and in how far smart educational tools can improve school results.

Industrial benefits The resulting three prototypes of this project are free to be used in schools. The large multimedia knowledge bases and the efficient semantic search engine are an attractive contribution to education, and coherent with today's pedagogy. Although our E-Librarian Service was implemented in educational prototypes, it can easily be used in different commercial areas like online helpdesks, or travel planers. We could imagine that clients requiring some help, e.g., with their Internet connection could contact a "virtual online help desk", and express questions in natural language. The expert system would understand the sense of the customer's question, and could propose a short but pertinent answer.

1.4 Organization of the Document

The thesis is structured as follows. After this brief introduction, the **first part** will describe the context of the research work. Chapter 2 will provide an introduction to the Semantic Web, its architecture and technologies. Chapter 3 will give an overview of Description Logics, the formalism our E-Librarian Service relies on. Chapter 4 will describe state-of-the art and related projects.

The technical contributions of this research work will be described in the **second part** of the thesis. Chapter 5 will motivate the ontological approach of our E-Librarian Service. Here, we will describe the used ontologies, as well as the semantic annotation of the documents in the knowledge base. Chapter 6 will explain the natural language interface. Three explored strategies will be described before focusing on the adopted one. Chapter 7 will present two explored retrieval strategies. Then, the adopted method relying on a modified version of the *concept covering problem* in Description Logics, will be presented. The chapter will conclude with an illustrating example of the retrieval algorithm. Chapter 9 will show the different prototypes that we developed: CHESt, MatES, and the Lecture Butler's E-Librarian Service. In the same chapter, a more detailed view about the architecture of our E-Librarian Service will be given, as well as technical details about its implementation. Chapter 8 will report on two benchmark tests that were carried out to evaluate the performance of our prototypes, and that confirmed their efficiency.

The **third part** of the thesis will describe the E-Librarian Service from the perspective of an educational tool. Chapter 11 will report on an experiment made in school, where the students' liking of such a "virtual teacher" was tested. The objective of that experiment was to test the users' acceptance to use such an interactive tool, and whether they accept to enter complete questions instead of keywords only. A more elaborated prototype of our E-Librarian Service was used in the experiment that will be described in chapter 12. Hence, the objective was to investigate if such a tool can be used efficiently in educational environment, and to find out its influence on the students' school results.

The **fourth part** of the thesis will conclude the research work. Chapter 13 will conclude the thesis with a brief summary of achieved results. Chapter 15 will summarize some future work, and will give details about automatically generating the semantic annotation for new documents in the knowledge base, and improving the search results by user feedback.

A list of the abbreviations and references used in the thesis is provided in the appendix.

Part I

Preliminaries

Chapter 2

Semantic Web

Contents

The Semantic Web is an evolving extension of the World Wide Web in which Web content can be expressed not only in natural language, but also in a form that can be read and "understood" by software agents, thus permitting them to find, share, and integrate information more easily. It derives from Tim Berners-Lee's vision of the Web as a universal medium for data, information, and knowledge exchange.

At its core, the Semantic Web comprises a philosophy, a set of design principles, collaborative working groups, and a variety of enabling technologies. Some elements of the Semantic Web are expressed as prospective future possibilities that have yet to be implemented or realized. Other elements are expressed in formal specifications, like the *Resource Description Framework* (RDF), and the *Web Ontology Language* (OWL), all of which are intended to provide a formal description of concepts, terms, and relationships within a given knowledge domain.

This chapter is organized as follows. Section 2.1 will give an introduction to the vision of the Semantic Web as seen by Tim Berners-Lee. The architecture of the Semantic Web will be depicted in section 2.2. An overview about ontologies in computer science will be given in section 2.3.

2.1 What is the Semantic Web?

There are a lot of different expressions and buzzwords about the evolving extension of the World Wide Web (WWW), e.g., “Web of the Next Generation”, “Semantic Web”, “Web 2.0“, ”Web 3.0”, “Social Web”, etc. The aim of this section is to focus on the scientific dimension of the WWW and its extensions as seen by the World Wide Web Consortium (W3C). We will briefly present some alternative evolutionary paths of the WWW. An exhaustive description is given in [MS04a].

2.1.1 The Vision of the Semantic Web

The WWW is a place where there is a huge amount of information. In that Web, machines are charged with the presentation of the information, which is a relatively simple task, and people must do the linking and interpreting, which is a much harder task, e.g., to find the information they are looking for. The obvious question is: “Why not get computers to do more of the hard work?”. That statement is representative for the discussion toward a new Web that was popularized by Tim Berners-Lee under the name of “Semantic Web” (SW) [BLHL01].

> The Semantic Web will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users. [...]
>
> The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation. [...] In the near future, these developments will usher in significant new functionality as machines become much better able to process and “understand” the data that they merely display at present.

The SW[1] is about two things. First, it is about common formats for integration and combination of data drawn from heterogeneous sources, whereas on the original Web mainly concentrated on the interchange of documents. Thus, the SW is often presented as being one huge database or a set of distributed databases. Secondly, it is also about language for recording how the data relates to real world objects. That allows a person or a machine to start off in one database, and then move through an unending set of databases, which are connected not by wires, but by the fact that they are about the same thing.

2.1.2 Other Perceptions

There is considerable debate about what the Web n.0 terms actually mean. It is suggested by many that such terms are just some buzzword, while the contrary view is that it is an evolutionary path for the WWW.

The phrase “Web 2.0” was coined in 2003 by O’Reilly Media[2]. Earlier, it was employed as a synonym for “Semantic Web”. Given the lack of set standards as to what Web 2.0 actually means, implies or requires, the term can mean radically different things to different people. In alluding to the version-numbers that commonly designate software upgrades, the phrase “Web 2.0” may hint at an improved form of the WWW. Advocates of the concept suggest that technologies such as Web blogs, social bookmarking, Wikis, podcasts, RSS feeds (and other forms of many-to-many publishing), social software, Web APIs, Web standards, and online Web services imply a significant change in Web usage. Sometimes, the “Web 2.0” is also used as synonym for “Social Web”.

[1] `http://www.w3.org/2001/sw/`
[2] `http://radar.oreilly.com/archives/2006/05/controversy_about_our_web_20_s.html`

The term "Web 3.0" first appeared prominently in early 2006 in a blog article by Jeffrey Zeldman[3]. It has been coined with different meanings to describe the evolution of Web usage and interaction along several separate paths. These include transforming the WWW into a database, a move towards making content accessible by multiple non-browser applications, the leveraging of artificial intelligence technologies and the SW, and three dimensional interaction and collaboration.

2.2 Architecture

The SW is composed of different layers, each having specified functions. Currently, there is an ongoing debate about the SW's architecture[4,5]. However, in this section we will only be able to give an uncomplete and subjective overview of different layers and technologies (see figure 2.1).

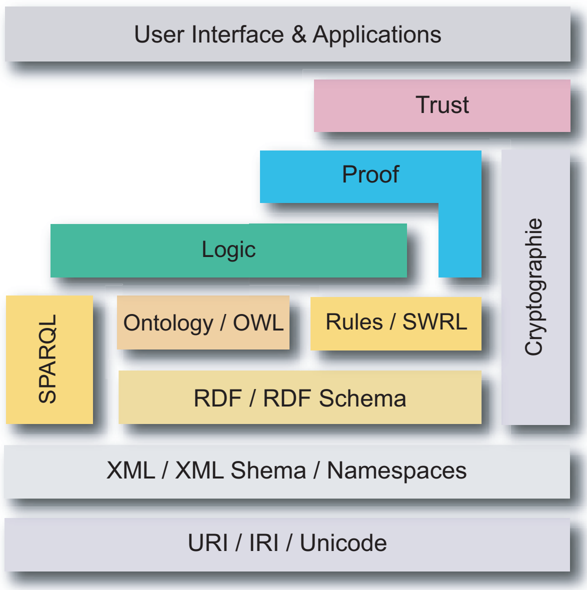


Figure 2.1: Semantic Web architecture, the so called "Layer Cake".

2.2.1 Unified Resource Identifier (URI)

The SW is based on one Web naming and addressing technology: *Uniform Resource Identifier*[6] (URI). URIs are short ASCII-strings that identify resources. Everything — including the real world — can be addressed and identified via URIs. URIs make resources available under a variety of naming schemes and access methods such as HTTP, FTP, and Internet mail addressable in the same simple way. It is an extensible technology; there are a number of existing addressing schemes, and more may be incorporated over time. Hence, a URI can be the address of a Web page (URL), or the ISBN of a book somewhere in a library.

Internationalized Resource Identifiers (IRIs) are a new protocol element, a complement to URIs. An IRI is a sequence of characters from the *Universal Character Set* (Unicode). There is a mapping

[3] http://www.alistapart.com/articles/web3point0
[4] http://www.w3.org/2007/Talks/0130-sb-W3CTechSemWeb
[5] http://www.w3.org/2004/Talks/0412-RDF-functions/Overview.html
[6] http://www.w3.org/Addressing/

from IRIs to URIs, which means that IRIs can be used instead of URIs where appropriate to identify resources.

2.2.2 Extensible Markup Language (XML)

Extensible Markup Language[7] (XML) is a simple, very flexible text format derived from *Standard Generalized Markup Language*[8] (SGML). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere.

XML Schema[9] (XMLS) express shared vocabularies and allow machines to carry out rules made by people. XMLS provides a means for defining the structure, content and syntax of XML documents.

2.2.3 Resource Description Framework (RDF)

Resource Description Framework[10] (RDF) is a specification of the W3C originally designed as a metadata model, but which has come to be used as a general method of modeling information through a variety of syntax formats. RDF is based upon the idea of making statements about resources in the form of "subject-predicate-object" expressions, called "triples" in RDF terminology. The subject denotes the resource, and the predicate denotes traits or aspects of the resource, and expresses a relationship between the subject and the object. This mechanism for describing resources is a major component of the Semantic Web activity [Hje01, Pow03].

SPARQL[11] is an RDF query language that stands for *SPARQL Protocol and RDF Query Language*. It allows for a query to consist of triple patterns, conjunctions, disjunctions, and optional patterns. Several implementations for multiple programming languages exist.

RDF Schema[12] (RDFS) is an extensible knowledge representation language, providing basic elements for the description of ontologies, otherwise called RDF vocabularies, intended to structure RDF resources.

An illustration is given in figure 2.2. It shows the statement "Serge is Magali's husband" as RDF-graph. The triple is composed of the resource Serge, the property isHusbandOf, and the value Magali, where Magali is a resource and not a literal. The RDFS serialization (code on the left-hand side) defines the two classes Man and Woman, both subclasses of Human, as well as the property isHusbandOf of the class Man. The `domain` specifies the class the property belongs to, and the `range` is used to specify the class the property can reference as values. The RDF serialization (code on the right-hand side) declares two instances; Magali as an individual of the class Woman, and Serge as an individual of the class Man. The value of the property isHusbandOf of the individual Serge is defined as being the individual Magali.

A powerful mechanism in RDF is *reification* that allows to make RDF statements about RDF triples. Thus, in the above example one could express the following: "Mike says that Serge is the husband of Magali". Here, Mike is the subject of the reification triple. However, while RDF provides this reification vocabulary, care is needed in using it, because it is easy to imagine that the vocabulary defines some things that are not actually defined[13].

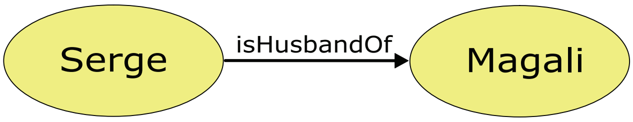
[7] `http://www.w3.org/XML/`
[8] `http://www.w3.org/MarkUp/SGML/`
[9] `http://www.w3.org/XML/Schema`
[10] `http://www.w3.org/RDF/`
[11] `http://www.w3.org/TR/rdf-sparql-query/`
[12] `http://www.w3.org/TR/rdf-schema/`
[13] `http://www.w3.org/TR/rdf-primer/#reification`

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:demo="http://www.linckels.lu/demo">

<rdfs:Class rdf:about="Woman">
  <rdfs:subClassOf rdf:resource="Human" />
</rdfs:Class>

<rdfs:Class rdf:about="Man">
  <rdfs:subClassOf rdf:resource="Human" />
</rdfs:Class>

<rdf:Property rdf:about="isHusbandOf">
  <rdfs:domain rdf:resource="Man"/>
  <rdfs:range rdf:resource="Woman"/>
</rdf:Property>
</rdf:RDF>
```




```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:demo="http://www.linckels.lu/demo">

<demo:Magali rdf:parseType="Woman"/>

<demo:Serge rdf:parseType="Man">
  <demo:isHusbandOf rdf:resource="Magali" />
</demo:Serge>

</rdf:RDF>
```

Figure 2.2: Illustration of an RDF triple; graphical representation (top, right hand-side), RDFS serialization (left hand-side), and RDF serialization (right hand-side).

2.2.4 Web Ontology Language (OWL) and Semantic Web Rule Language (SWRL)

In both computer science and information science, an ontology is a data model that represents a set of concepts within a domain, and the relationships between those concepts (see section 2.3). It is used to reason about the objects within that domain. Ontologies are used in the SW as a form of knowledge representation about the world or some part of it.

Web Ontology Language[14] (OWL) is a language for defining and instantiating Web ontologies [Lac05]. An OWL ontology may include descriptions of classes along with their related properties and instances. OWL is designed for use by applications that need to process the content of information instead of just presenting information to humans. It facilitates greater machine interpretability of Web content than that supported by XML(S) and RDF(S) by providing additional vocabulary along with a formal semantics.

OWL has three sublanguages, sometimes also referred to as “flavors” or “species”: OWL Lite, OWL DL, and OWL Full. These three increasingly expressive sublanguages are designed for use by specific communities of implementers and users.

- *OWL Lite* supports those users primarily needing a classification hierarchy and simple constraints. Hence, while it supports cardinality constraints, it only permits cardinality values of 0 or 1. It should be simpler to provide tool support for OWL Lite than its more expressive relatives, and OWL Lite provides a quick migration path for thesauri and other taxonomies. OWL Lite also has a lower formal complexity than OWL DL.

- *OWL DL* supports those users who want the maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computed) and decidability (all computations will finish in finite time). OWL DL includes all OWL language constructs, but they can be used only under certain restrictions (e.g., no reification). OWL DL is so named

[14] http://www.w3.org/2004/OWL/

due to its correspondence with Description Logics (see chapter 3), a field of research that has studied the logics that form the formal foundation of OWL.

- *OWL Full* is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. Thus, in OWL Full a class can be treated simultaneously as a collection of individuals, and as an individual in its own right. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full.

The relation between DL and OWL, as well as an illustration of a OWL DL serialization are depicted in section 3.4.

Semantic Web Rule Language[15] (SWRL) is a proposal for a SW rule-language, combining sublanguages of the OWL (Lite and DL) with those of the *Rule Markup Language*[16]. Other rule languages are: *Description Logic Programs*[17] (DLP), and *Rule Interchange Format*[18] (RIF). Rules are Horn-clauses of the form of an implication between an antecedent (body) and consequent (head). The intended meaning can be read as: whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold. An interesting application for query answering was published in [MSS05].

2.2.5 Logic, Proof, Trust

If the SW is indeed to become a global database, and if its development is evolutionary and distributed, then there are issues of accessibility, trust and credibility[19]. Not all data sources will have universal access, so there needs to be a robust and extensible security model. Not all data sources will be equally reliable. If instead of just returning an answer to a query, a SW application could also attach a proof of how that answer was derived, then the querying application could potentially do some reasoning about how "believable" that fact is. At the very least, derived facts could be attributed to a source, and over time applications could be developed which rate sources as to their integrity. These upper layers of the stack are the least researched, and present some of the most difficult technical challenges faced by the SW venture, see e.g., [OAKS04, VdSA06, BDF+06].

XML Signatures and *XML Encryption* are examples of already existing and underlying technologies. The XML Signature[20] is a method of associating a key with referenced data (octets); it does not normatively specify how keys are associated with persons or institutions, nor the meaning of the data being referenced and signed. XML Encryption[21] is a specification that defines how to encrypt the content of an XML element. It encompasses the encryption of any kind of data, including the encryption of XML. What makes it XML Encryption is that an XML element (either an `EncryptedData` or `EncryptedKey` element) contains or refers to the cipher text, keying information, and algorithms.

[15] `http://www.w3.org/Submission/SWRL/`
[16] `http://www.ruleml.org/`
[17] `http://kaon.semanticweb.org/alphaworld/dlp`
[18] `http://www.w3.org/2005/rules/`
[19] `http://sites.wiwiss.fu-berlin.de/suhl/bizer/SWTSGuide/`
[20] `http://www.w3.org/TR/xmldsig-core/`
[21] `http://www.w3.org/TR/xmlenc-core/`

2.3 Ontologies

The term ontology has its origin in philosophy, where it is the name of one fundamental branch of metaphysics concerned with analyzing various types or modes of existence, often with special attention to the relations between particulars and universals, between intrinsic and extrinsic properties, and between essence and existence.

In this section, we will focus on ontologies in computer science. We will start with a description of the structure of ontologies, and will introduce some ontology languages. Then, we will describe the differences between *upper ontologies* and *domain ontologies*, whilst giving some examples of famous ontologies.

2.3.1 Ontology Structure

In computer science, "*an ontology is a specification of a conceptualization*" [Gru93], or like paraphrased, "*ontologies are formal and consensual specifications of conceptualizations that provide a shared understanding of a domain, an understanding that can be communicated across people and application systems*" [Fen04]. What ontology has in common in both computer science and philosophy is the representation of entities (individuals), ideas (classes), along with their properties (attributes) and relations. We briefly describe these different components.

2.3.1.1 Individuals

Individuals (instances) are the basic, "ground level" components of an ontology. The individuals in an ontology may include concrete objects such as people, animals, tables, automobiles, molecules, and planets, as well as abstract individuals such as numbers and words. Strictly speaking, an ontology needs not include any individuals, but one of the general purposes of an ontology is to provide a means of classifying individuals, even if those individuals are not explicitly part of the ontology.

2.3.1.2 Classes

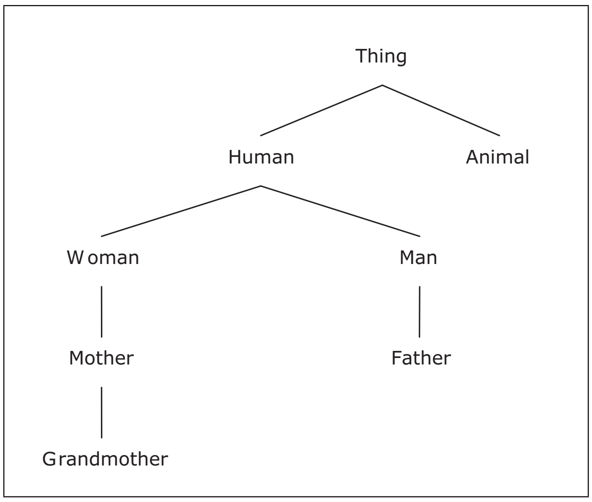


Figure 2.3: Example of a class hierarchy (taxonomy).

Classes (concepts) are abstract groups, sets, or collections of objects (individuals and classes). They may contain individuals, other classes, or a combination of both. Importantly, a class can subsume or be subsumed by other classes. Thus, Human subsumes Woman, since (necessarily) anything that is a member of the latter class is a member of the former. The subsumption relation is used to create a hierarchy of classes (see figure 2.3), typically with the most general class like Thing at the top, and very specific classes at the bottom.

2.3.1.3 Attributes

Objects (individuals and classes) in the ontology can be described by assigning attributes (properties) to them. Attributes over classes have a name, and attributes over individuals have in addition a value that is used to store information that is specific to the individual it is attached to. For instance, the class Mother has the attribute hasChild, and the individual Mother(Mathilde) has the attribute hasChild with the value: {Mike, Serge}. The value of an attribute can be a literal, or a complex data type like in this example; the value of hasChild is a set of values, not just a single value.

2.3.1.4 Relationships

An important use of attributes is to describe the relationships between objects in the ontology. Typically a relation is an attribute whose value is another object in the ontology, e.g., a successor in the hierarchy. The most important type of relation is the *subsumption relation*, commonly called "is-superclass-of", the converse of "is-a", "is-subtype-of", or "is-subclass-of". This defines which objects are members of classes of objects.

The addition of the is-a relationships has created a hierarchical taxonomy; a tree-like structure that clearly depicts how objects relate to one another (see figure 2.3). In such a structure, each object is the "child" (subclass) of a "parent class" (superclass). Some languages restrict the is-a relationship to one parent for all nodes.

Other relations can be defined, e.g., the relation isHusbandOf from the class Man to the class Woman, or the reflexive relation hasChild from the class Human to itself. In the first example, the emerging structure is a directed acyclic graph, whereas in the second example the emerging structure would be a directed cyclic graph.

2.3.2 Ontology Language

Beside the structural dimension of an ontology, i.e., the classes and relations between classes, an ontology uses a common language to formalize its specifications and conceptualizations; OWL is the most common used in the SW (see section 2.2.4). Others are ancestors of OWL like OIL[22], DAML[23], DAML+OIL[24], or more specific languages like the *Knowledge Interchange Format* (KIF)[25] that was created to serve as a syntax for first-order logic, and *CycL*[26] a declarative language used in the Cyc project.

[22] http://www.ontoknowledge.org/oil/
[23] http://www.daml.org/
[24] http://www.daml.org/2001/03/daml+oil-index
[25] http://suo.ieee.org/SUO/KIF/suo-kif.html
[26] http://www.cyc.com/cycdoc/ref/cycl-syntax.html

2.3.3 Upper- and Domain Ontologies

An **upper ontology** (or world ontology) is a model of the common objects that are generally applicable across a wide range of domain ontologies. It contains a core glossary in whose terms objects in a set of domains can be described. There are several standardized upper ontologies available, e.g.:

- The *Dublin Core*[27] metadata element set is a standard for cross-domain information resource description. In other words, it provides a simple and standardized set of conventions for describing things online in ways that make them easier to find. Dublin Core is widely used to describe digital materials such as video, sound, image, text, and composite media like Web pages.

- The *General Formal Ontology*[28] (GFO) is an upper ontology integrating processes and objects. GFO provides a framework for building custom, domain-specific ontologies.

- *OpenCyc*[29] includes hundreds of thousands of terms along with millions of assertions relating the terms to each other. One stated goal is that of providing a completely free and unrestricted semantic vocabulary for use in the SW. The OpenCyc taxonomy is available in OWL.

- *Suggested Upper Merged Ontology* (SUMO)[30] was developed within the IEEE Standard Upper Ontology Working Group. The goal is to develop a standard ontology that will promote data interoperability, information search and retrieval, automated inferencing, and natural language processing.

- *PROTON*[31] is a basic upper-level ontology which contains about 300 classes and 100 properties, providing coverage of the general concepts necessary for a wide range of tasks, including semantic annotation, indexing, and retrieval of documents.

A **domain ontology** models a specific domain, or part of the world. It represents the particular meanings of terms as they apply to that domain. For example, the word "golf" has many different meanings. An ontology about the domain of automobiles would model the "kind of car" meaning of the word, an ontology about the domain of sports would model the "kind of game" meaning to the word, while an ontology about the domain of geography would model the "geographical location" meanings. Here are some examples of domain ontologies:

- One of the most cited ontologies is the "wine ontology"[32]; it is about the most appropriate combination of wine and meals.

- The "soccer ontology"[33] describes most concepts that are specific to soccer: players, rules, field, supporters, actions, etc. It is used to annotate videos in order to produce personalized summary of soccer matches.

[27] `http://dublincore.org/`
[28] `http://www.onto-med.de/en/theories/gfo/`
[29] `http://www.opencyc.org/`
[30] `http://www.ontologyportal.org/`
[31] `http://proton.semanticweb.org/`
[32] `http://www.w3.org/TR/owl-guide/wine.rdf`
[33] `http://www.daml.org/ontologies/273`

- An ontology library for lung pathology[34] is maintained by the FU-Berlin. The aim of the project "A Semantic Web for Pathology" is to realize a SW based retrieval system for the domain of lung pathology. For this purpose the pathology data is annotated with semantic references, and the textual pathology reports are used as descriptions of what the associated images represent.

Since domain ontologies represent concepts in very specific and often eclectic ways, they are often incompatible. As systems that rely on domain ontologies expand, they often need to merge domain ontologies into a more general representation. This presents a challenge to the ontology engineer. Different ontologies in the same domain can also arise due to different perceptions of the domain based on cultural background, education, ideology, or because a different representation language was chosen.

Research in ontological engineering — see, e.g., [GPCGFL03, SS04, AvH04] for an overview — aims to create specialized domain ontologies for every imaginable topic, rather than develop large upper ontologies. Therefore, a lot of efforts are put into the development of tools for creating and maintaining ontologies, e.g., Protégé[35] [KFNM04]. Also, operations over ontologies is currently an area of research, like semantically comparing related ontologies [SMMS02], mappings between ontologies [BSZ04, KBHS04], ontology matching [ES07] and algebras for ontologies [Luc06].

[34] `http://swpatho.ag-nbi.de/english/ontologies.html`
[35] `http://protege.stanford.edu/`

Chapter 3

Description Logics

Contents

This chapter will provide an introduction to Description Logics as a formal language for representing knowledge and reasoning about it. Section 3.1 will give a short overview of the ideas underlying Description Logics. It will introduce syntax and semantics, covering the basic constructors that are used in systems or have been introduced in the literature, and the way these constructors can be used to build knowledge bases. Section 3.2 will depict the main feature of Description Logics and Description Logics-based systems: inference services. We will group them into standard- and non-standard inferences. Two reasoning algorithms will be presented in section 3.3: structural subsumption and tableau algorithms. Finally, section 3.4 will give an overview of the intrinsic relation between OWL and Description Logics.

3.1 Basic Notions

Description Logics[1] (DLs) [BCM+03] denote a family of knowledge representation formalisms that allow to represent the terminological knowledge of an application domain in a structured and well-defined way. A DL-knowledge base comprises two components: the TBox and the ABox. The TBox introduces the terminology, i.e., the vocabulary of an application domain, while the ABox contains assertions about named individuals in terms of this vocabulary. Before elaborating on these subjects, we will introduce the notions of concept descriptions and interpretations.

3.1.1 Concept Descriptions

In DLs, the conceptual knowledge of an application domain is represented in terms of *concepts* (unary predicates) such as Human and Woman, and *roles* (binary predicates) such as hasChild. Concepts denote sets of individuals and roles denote binary relations between individuals. Based on basic concept and role names, complex concept descriptions are built inductively using concept constructors.

C,D	$\rightarrow$	A	atomic concept
		$\top$	universal (top) concept
		$\bot$	bottom concept
		$\neg A$	atomic negation
		$\sqcap$	intersection
		$\forall R.C$	value restriction
		$\exists R.\top$	limited existential quantification

Figure 3.1: Syntax rule in the language $\mathcal{AL}$.

The language $\mathcal{AL}$ (attributive language) [SSS91] is a minimal attributive language. Concept descriptions in $\mathcal{AL}$ are formed according to the syntax rule shown in figure 3.1. In that abstract notation, A and B denote atomic concepts, R denotes an atomic role, and C and D denote concept descriptions. Hence, the following $\mathcal{AL}$-concept description represents all women that have at least one human child, i.e., who are a mother:

$$\mathsf{Woman} \sqcap \exists \mathsf{hasChild}.\mathsf{Human}$$

The different DLs languages distinguish themselves by the kind of constructs they allow. An overview of frame based languages ($\mathcal{FL}$) and attributive languages ($\mathcal{AL}$) is shown in figure 3.2.

3.1.2 Interpretations

In order to define a formal semantics of $\mathcal{AL}$-concepts, we consider interpretations $\mathcal{I}$. An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty set $\Delta^{\mathcal{I}}$ (the domain of the interpretation) and an interpretation function $\cdot^{\mathcal{I}}$, which assigns to every atomic concept A a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to every atomic role R a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function is extended to concept descriptions by the following inductive definitions:

[1] `http://dl.kr.org/`

	A	$\top$	$\bot$	$\neg A$	$C \sqcap D$	$C \sqcup D$	$\neg C$	$\forall R.C$	$\exists R.\top$	$\exists R.C$	$\geq nR$	$\leq nR$	$= nR$
$\mathcal{ALN}$	×	×	×	×	×	×		×	×		×	×	×
$\mathcal{ALC}$	×	×	×	×	×	×	×	×	×				
$\mathcal{ALU}$	×	×	×	×	×	×		×	×				
$\mathcal{ALE}$	×	×	×	×	×			×	×	×			
$\mathcal{AL}$	×	×	×	×	×			×	×				
$\mathcal{FL}^-$	×	×			×			×	×				
$\mathcal{FL}_\bot$	×	×	×		×			×					
$\mathcal{FL}_0$	×	×			×			×					
$\mathcal{EL}$	×	×			×				×	×			
$\mathcal{L}$	×	×			×								

Figure 3.2: Overview of description languages and their concept constructors.

$$
\begin{aligned}
\top^\mathcal{I} &= \Delta^\mathcal{I} \\
\bot^\mathcal{I} &= \emptyset \\
(\neg A)^\mathcal{I} &= \Delta^\mathcal{I} \setminus A^\mathcal{I} \\
(\neg C)^\mathcal{I} &= \Delta^\mathcal{I} \setminus C^\mathcal{I} \\
(C \sqcap D)^\mathcal{I} &= C^\mathcal{I} \cap D^\mathcal{I} \\
(C \sqcup D)^\mathcal{I} &= C^\mathcal{I} \cup D^\mathcal{I} \\
(\forall r.C)^\mathcal{I} &= \{x \in \Delta^\mathcal{I} \mid \forall y : (x,y) \in r^\mathcal{I} \rightarrow y \in C^\mathcal{I}\} \\
(\exists r.\top)^\mathcal{I} &= \{x \in \Delta^\mathcal{I} \mid \exists y : (x,y) \in r^\mathcal{I}\} \\
(\exists r.C)^\mathcal{I} &= \{x \in \Delta^\mathcal{I} \mid \exists y : (x,y) \in r^\mathcal{I} \wedge y \in C^\mathcal{I}\} \\
(\geq n\ r)^\mathcal{I} &= \{x \in \Delta^\mathcal{I} \mid \sharp\{y \in \Delta^\mathcal{I} \mid (x,y) \in r^\mathcal{I}\} \geq n\} \\
(\leq n\ r)^\mathcal{I} &= \{x \in \Delta^\mathcal{I} \mid \sharp\{y \in \Delta^\mathcal{I} \mid (x,y) \in r^\mathcal{I}\} \leq n\}
\end{aligned}
$$

3.1.3 Terminologies (TBox)

We have seen how we can form complex descriptions of concepts to describe classes of objects. Now, we introduce terminological axioms, which make statements about how concepts or roles are related to each other. Terminologies are composed of terminological axioms which can be *definitions* and *inclusion assertions*.

3.1.3.1 Definitions

Definitions allow to give a meaningful name (*concept name* or *symbolic name*) to concept descriptions, e.g., to define that a mother is a woman that has at least one human child one can write:

$$\mathsf{Mother} \equiv \mathsf{Woman} \sqcap \exists \mathsf{hasChild}.\mathsf{Human}. \tag{3.1}$$

Here, Mother is the concept name that identifies the concept description (on the right-hand side of the equivalent symbol). Woman and Human are *atomic concepts*. If in a terminology an atomic concept appears only on the right-hand side of a concept description, then it is called a *primitive concept*, otherwise it is a *defined concept*.

A terminology is *cyclic* when its definitions are cyclic. In general, we define cycles in a terminology $\mathcal{T}$ as follows. Let A, B be atomic concepts occurring in $\mathcal{T}$. We say that A directly uses B in $\mathcal{T}$ if B appears on the right-hand side of the definition of A, and we call "uses" the transitive closure of the relation "directly uses". Then $\mathcal{T}$ contains a cycle iff there exists an atomic concept in $\mathcal{T}$ that uses itself. Otherwise, $\mathcal{T}$ is called acyclic. An example of an acyclic terminology is shown in figure 3.3.

$$
\begin{array}{lcl}
\mathsf{Woman} & \equiv & \mathsf{Human} \sqcap \mathsf{Female} \\
\mathsf{Man} & \equiv & \mathsf{Human} \sqcap \neg\mathsf{Woman} \\
\mathsf{Mother} & \equiv & \mathsf{Woman} \sqcap \exists\mathsf{hasChild}.\mathsf{Human} \\
\mathsf{Father} & \equiv & \mathsf{Man} \sqcap \exists\mathsf{hasChild}.\mathsf{Human} \\
\mathsf{Parent} & \equiv & \mathsf{Mother} \sqcup \mathsf{Father} \\
\mathsf{GrandMother} & \equiv & \mathsf{Mother} \sqcap \exists\mathsf{hasChild}.\mathsf{Parent}
\end{array}
$$

Figure 3.3: Example of a terminology w.r.t. to the taxonomy shown in figure 2.3.

A terminology that is acyclic can be *expanded*. This can be done through an iterative process over the definitions in $\mathcal{T}$ by replacing each occurrence of a concept name on the right-hand side of a definition with the concepts that it stands for. Since there is no cycle in the set of definitions, the process eventually stops and we end up with a terminology $\mathcal{T}'$ consisting solely of definitions of the form $A \equiv C'$, where C' contains only primitive concepts and no defined concepts. We call $\mathcal{T}'$ the expansion of $\mathcal{T}$. An example of an expanded terminology is shown in figure 3.4.

$$
\begin{array}{lcl}
\mathsf{Woman} & \equiv & \mathsf{Human} \sqcap \mathsf{Female} \\
\mathsf{Man} & \equiv & \mathsf{Human} \sqcap \neg\mathsf{Woman} \\
\mathsf{Mother} & \equiv & \mathsf{Human} \sqcap \mathsf{Female} \sqcap \exists\mathsf{hasChild}.\mathsf{Human} \\
\mathsf{Father} & \equiv & \mathsf{Human} \sqcap \neg(\mathsf{Human} \sqcap \mathsf{Female}) \sqcap \exists\mathsf{hasChild}.\mathsf{Human} \\
\mathsf{Parent} & \equiv & (\mathsf{Human} \sqcap \mathsf{Female} \sqcap \exists\mathsf{hasChild}.\mathsf{Human}) \sqcup \\
 & & (\mathsf{Human} \sqcap \neg(\mathsf{Human} \sqcap \mathsf{Female}) \sqcap \exists\mathsf{hasChild}.\mathsf{Human}) \\
\mathsf{GrandMother} & \equiv & \mathsf{Human} \sqcap \mathsf{Female} \sqcap \exists\mathsf{hasChild}.(\\
 & & (\mathsf{Human} \sqcap \mathsf{Female} \sqcap \exists\mathsf{hasChild}.\mathsf{Human}) \sqcup \\
 & & (\mathsf{Human} \sqcap \neg(\mathsf{Human} \sqcap \mathsf{Female}) \sqcap \exists\mathsf{hasChild}.\mathsf{Human})
\end{array}
$$

Figure 3.4: The expansion of the terminology from figure 3.3.

3.1.3.2 Inclusion Assertions

For certain concepts we may be unable to define them completely. In this case, we can still state necessary conditions for an individual to belong to a concept using an inclusion assertion. We call an inclusion assertion whose left-hand side is atomic a *specialization*. Thus, to express that a Woman is, among other things, a specialization of a human, one can use the inclusion assertion:

$$\mathsf{Woman} \sqsubseteq \mathsf{Human}. \tag{3.2}$$

If we also allow specializations in a terminology, then the terminology loses its definitorial impact, even if it is acyclic. A set of axioms $\mathcal{T}$ is a *generalized terminology* if the left-hand side of

Animal	$\sqsubseteq$	Thing
Human	$\sqsubseteq$	Thing
Parent	$\sqsubseteq$	Human
Woman	$\sqsubseteq$	Human
Man	$\sqsubseteq$	Human
Mother	$\sqsubseteq$	Woman
Father	$\sqsubseteq$	Man
GrandMother	$\sqsubseteq$	Mother

Figure 3.5: Example of generalized terminology w.r.t. to the taxonomy shown in figure 2.3.

each axiom is an atomic concept and for every atomic concept there is at most one axiom where it occurs on the left-hand side. An example of a generalized terminology is shown in figure 3.5.

We shall transform a generalized terminology $\mathcal{T}$ into a regular terminology $\overline{\mathcal{T}}$, containing definitions only, such that $\overline{\mathcal{T}}$ is equivalent to $\mathcal{T}$ in a sense that will be specified below. We obtain $\overline{\mathcal{T}}$ from $\mathcal{T}$ by choosing for every specialization $A \sqsubseteq C$ in $\mathcal{T}$ a new base symbol $\overline{A}$ — where $\overline{A}$ stands for all the qualities that distinguish A from C — and by replacing the specialization $A \sqsubseteq C$ with the definition $A \equiv \overline{A} \sqcap C$. The terminology $\overline{\mathcal{T}}$ is the *normalization* of $\mathcal{T}$. In this respect, if a TBox contains the specialization (3.2), then the normalization contains the definition:

$$\mathsf{Woman} \equiv \overline{\mathsf{Woman}} \sqcap \mathsf{Human},$$

where $\overline{\mathsf{Woman}}$ stands for all the qualities that distinguish a woman among humans.

3.1.4 World Descriptions (ABox)

The second component of a DL-knowledge base — in addition to the terminology (TBox) — is the world description or ABox. In the ABox, one introduces individuals by giving them names, and one asserts properties of these individuals. There are two kinds of assertions: *concept assertions* and *role assertions*. By a concept assertion, one states that a certain individual a belongs to a concept C, written $C(a)$. By a role assertion, one states that an individual c is a filler of the role R for an individual b, written $R(b, c)$. For instance, to denote that Mathilde is a mother, and Serge a man who is the son of Mathilde, we write:

$$\mathsf{Mother(Mathilde)} \qquad \mathsf{Man(Serge)} \qquad \mathsf{hasChild(Mathilde,Serge)} \tag{3.3}$$

where the first two are concept assertions, and the third is a role assertion.

We give a semantics to ABoxes by extending interpretations to individual names. From now on, an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ not only maps atomic concepts and roles to sets and relations, but in addition maps each individual name a to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. We assume that distinct individual names denote distinct objects. A complete example is shown in appendix B.

3.2 Inferences

The main feature of DLs and DL-based knowledge representation systems are inference services, which allow to derive implicit knowledge from the knowledge explicitly stored in the knowledge base. We will give here an overview of the most important ones, typically divided into standard- and non-standard inferences.

3.2.1 Standard Inferences

"Classical" reasoning services like satisfiability check, subsumption check, etc. are called *standard inferences*. We briefly recall the most important ones; the first three are for concepts (C, D), the fourth is for TBoxes $(\mathcal{T})$, and the last two are for ABoxes $(\mathcal{A})$.

3.2.1.1 Concept Satisfiability (concepts)

A concept may be neither *true* nor *false* in an interpretation, it may simply be satisfiable. Testing if a concept is satisfiable means, testing if its interpretation is not an empty set. For example, C is satisfiable in $\mathcal{I}$ if $C^{\mathcal{I}} \neq \emptyset$. In that case we say that $\mathcal{I}$ is a model of C.

Checking satisfiability of concepts is a key inference. A number of other important inferences for concepts can be reduced to (un)satisfiability.

3.2.1.2 Subsumption (concepts)

Subsumption allows to compute subconcept/superconcept relationships. For example, $C \sqsubseteq D$ means that the concept D (the *subsumer*) is more general than the concept C (the *subsumee*). One says C is subsumed by D, or D subsumes C. $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all interpretations $\mathcal{I}$. One can write $C \sqsubseteq_{\mathcal{T}} D$ or $\mathcal{T} \models C \sqsubseteq D$ which says that C is subsumed by D w.r.t. $\mathcal{T}$.

For DLs that allow full negation (like $\mathcal{ALC}$) subsumption can be reduced to unsatisfiability since $C \sqsubseteq_{\mathcal{T}} D$ iff $C \sqcap \neg D \equiv_{\mathcal{T}} \bot$. Also, satisfiability can be reduced to subsumption since C is satisfiable in $\mathcal{I}$ if $C \not\sqsubseteq \bot$ or $\mathcal{T} \not\models C \equiv \bot$

3.2.1.3 Equivalence and Disjointness (concepts)

Two concepts C, D are equivalent w.r.t. $\mathcal{T}$, written $C \equiv_{\mathcal{T}} D$ or $\mathcal{T} \models C \equiv D$, if $C^{\mathcal{I}} = D^{\mathcal{I}}$ for all interpretations $\mathcal{I}$. In the same sense, two concepts are disjoint w.r.t. $\mathcal{T}$, written $C \not\equiv_{\mathcal{T}} D$ or $\mathcal{T} \not\models C \equiv D$, if $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ for all interpretations $\mathcal{I}$.

Testing equivalence can be reduced to subsumption checking because $C \equiv_{\mathcal{T}} D$ iff $C \sqsubseteq_{\mathcal{T}} D$ and $D \sqsubseteq_{\mathcal{T}} C$. Also, subsumption can be reduced to equivalence checking since $C \sqsubseteq_{\mathcal{T}} D$ iff $C \equiv_{\mathcal{T}} C \sqcap D$.

3.2.1.4 Reasoning over a TBox

In applications, concepts usually come in the context of a TBox. However, to develop reasoning procedures it is conceptually easier to abstract from the TBox or, what amounts to the same, to assume that it is empty. It was shown in [BCM+03] that, if $\mathcal{T}$ is an acyclic TBox, we can always reduce reasoning problems w.r.t. $\mathcal{T}$ to problems w.r.t. the empty TBox.

As we have already seen in section 3.1.3.1, $\mathcal{T}$ is equivalent to its expansion $\mathcal{T}'$. Let us recall that in the expansion every definition is of the form $A \equiv D$ such that D contains only primitive concepts, but no concept names.

We can readily deduce a number of facts about expansions. Since the expansion C' is obtained from C by replacing names with descriptions in such a way that both are interpreted in the same way in any model of $\mathcal{T}$, it follows that: $C \equiv_{\mathcal{T}} C'$, $D \equiv_{\mathcal{T}} D'$, $C \sqsubseteq_{\mathcal{T}} D$ iff $C' \sqsubseteq_{\mathcal{T}} D'$, and $C \equiv_{\mathcal{T}} D$ iff $C' \equiv_{\mathcal{T}} D'$. Also, C and D are disjoint w.r.t. $\mathcal{T}$ iff C' and D' are disjoint.

Similar assertions can be made of generalized terminologies. For illustration, suppose the following TBox that is composed only of one inclusion assertion: $\mathcal{T} = \{\mathsf{Woman} \sqsubseteq \mathsf{Human}\}$, then:

- $\mathsf{Woman} \sqsubseteq_{\mathcal{T}} \mathsf{Human}$
  holds because the subsumption is tested w.r.t. to $\mathcal{T}$,

- $\mathsf{Woman} \sqsubseteq \mathsf{Human}$
  does not hold because the subsumption is tested over an empty TBox,

- $\mathsf{Woman} \sqcap \mathsf{Human} \sqsubseteq \mathsf{Human}$
  holds because the subsumption is tested over an extended inclusion assertion.

3.2.1.5 Consistency check (ABox)

An ABox $\mathcal{A}$ is consistent (written $\mathcal{A} \not\models$) if no assertion is contradictory to the TBox, which means that all assertions can be satisfiable simultaneously. For example, consider the TBox shown in figure 3.3, then Mother(Mathilde) and Father(Mathilde) result in an inconsistency because the TBox specifies that the concepts Mother and Father are disjoint (a mother is a woman and a father is a man, but a woman cannot be a man). Thus, an individual cannot belong to both concepts simultaneously.

3.2.1.6 Instance Checking (ABox)

We say that an assertion α is *entailed* by the ABox $\mathcal{A}$, written $\mathcal{A} \models \alpha$, if every interpretation that satisfies $\mathcal{A}$, i.e., every model of $\mathcal{A}$, also satisfies α. For example, testing if an individual a is an instance of a concept C is written $\mathcal{A} \models C(a)$.

Instance checking can be reduced to consistency: $\mathcal{A} \models C(a) \equiv \mathcal{A} \cup \{\neg C(a)\}$ is inconsistent. For instance, if it was stated in the ABox $\mathcal{A}$ that Mother(Mathilde), then $\mathcal{A} \models$ Mother(Mathilde) is satisfied.

An extended form of instance checking is the so called *retrieval problem.* It is stated as follows: given an ABox $\mathcal{A}$ and a concept C, find all individuals a such that $\mathcal{A} \models C(a)$.

3.2.2 Non-Standard Inferences

Although standard inferences help structuring a knowledge base, e.g., by automatically building a concept hierarchy, they are not sufficient when it comes to (automatically) generating new concept descriptions from given ones. They also fail if concepts are specified using different vocabularies (i.e., sets of concept names and role names) or if they are described on different levels of abstraction. Altogether, it has turned out that for building and maintaining large DL knowledge bases besides the standard inferences, additional so-called non-standard inferences are required. Non-standard inferences are a group of relatively new inference services, which provide reasoning support for the building, maintaining, and deployment of DL knowledge-bases. So far, non-standard inferences are available for very expressive DLs but with no computational guarantees.

3.2.2.1 Most Specific Concept (mcs)

Intuitively, the most specific concept [Neb90] of individuals described in an ABox is a concept description that represents all the properties of the individuals including the concept assertions they occur in and their relationship to other individuals.

Definition 1 (most specific concept) *Let $\mathcal{A}$ be an $\mathcal{L}$-ABox and $a_1, ..., a_k$ be individuals of $\mathcal{A}$. Then, C is a most specific concept (msc) of $a_1, ..., a_k$ iff:*

- $\mathcal{A} \models C(a_i), i \in [1..k]$*, and*

- *C is the most specific concept with this property, i.e., for all $\mathcal{L}$-concept descriptions D, if $\mathcal{A} \models D(a_i), i \in [1..k]$, then $C \sqsubseteq D$.*

Consider the TBox shown in figure 3.3, and the ABox shown in (3.3), then msc(Mathilde,Serge) = Human.

3.2.2.2 Least Common Subsumer (lcs)

The least common subsumer (lcs) [BKM99] stands for the least concept description (w.r.t. subsumption) that subsumes a given set of concept descriptions.

Definition 2 (least common subsumer) *Let $C_1, ..., C_k$ be $\mathcal{L}$-concept descriptions. The $\mathcal{L}$-concept description C is a least common subsumer (lcs) of $C_1, ..., C_k$ iff:*

- *$C_i \sqsubseteq C$ for all $i = 1, ..k$, and*
- *C is the most specific concept with this property, i.e., for every $\mathcal{L}$-concept description E, if $C_i \sqsubseteq E$ for all $i = 1, ..., k$, then $C \sqsubseteq E$.*

In the TBox shown in figure 3.3, lcs(Mother,Father) = Parent, and lcs(Mother,Animal) = Thing.

3.2.2.3 Difference Operation

The difference operation [Tee94] allows to remove from a given concept description all the information contained in another concept description.

Definition 3 (semantic difference) *Let $C, D \in \mathcal{L}$ be two descriptions with $C \sqsubseteq D$. The difference is defined by:*

$$C - D := \max_{\sqsubseteq}\{B \in \mathcal{L} | B \sqcap D \equiv C\}.$$

First, every description B in the result contains enough information to yield the information in C if added to D, i.e., it contains all information from C which is missing in D. Secondly, B is **maximally** general, i.e., it does not contain any additional unnecessary information.

Consider the TBox shown in figure 3.3, then Father $-$ Man $= \exists$hasChild.Human. Here are some more examples:

C	D	$C - D$
$A_1 \sqcap A_2 \sqcap A_3 \sqcap A_4$	$A_2 \sqcap A_3$	$A_1 \sqcap A_4$
A	$\top$	A, because $A \sqcap \top \equiv A$
$A_1 \sqcap A_2 \sqcap A_3$	B	not possible because $C \not\sqsubseteq D$

In some DLs, the difference may contain descriptions which are not semantically equivalent. Teege [Tee94] defines necessary conditions for a DL to have a semantically unique difference. Those DLs are said with *structural subsumption*, see section 3.3.1.

The above definition of semantic difference requires that the second argument subsumes the first one. However, the semantic difference $C - D$ between two incomparable descriptions C and D can be given by computing the least common subsumer of C and D:

$$C - D = C - lcs(C, D).$$

For illustration, suppose that $(A_1 \sqcap A_2 \sqcap A_3) \not\sqsubseteq B$, then:

$$\begin{aligned}(A_1 \sqcap A_2 \sqcap A_3) - B &= (A_1 \sqcap A_2 \sqcap A_3) - lcs(A_1 \sqcap A_2 \sqcap A_3, B)\\ &= (A_1 \sqcap A_2 \sqcap A_3) - \top\\ &= A_1 \sqcap A_2 \sqcap A_3\end{aligned}$$

A new definition of difference operator was given in [BKT02]; the *syntactic difference.* The only difference to Teege's difference operator is that the minimum w.r.t. $\prec_d$ is used instead of the maximum w.r.t. a syntactic order $\sqsubseteq$. We describe it in appendix C.

3.2.2.4 Concept Rewriting

Given a concept expressed in a source language, concept rewriting aims to find a concept, possibly expressed in a target language, which is related to the given concept according to equivalence, subsumption, or some other relation. Concept rewriting can be applied to the translation of concepts from one knowledge base to another, or in the reformulation of concepts during the process of knowledge base construction and maintenance [BKM00].

For example, if $\mathcal{T}$ contains the definition $\mathsf{Parent} \equiv \mathsf{Human} \sqcap \exists \mathsf{hasChild.Human}$, then the concept description $\mathsf{Human} \sqcap \exists \mathsf{hasChild.(Human} \sqcap \exists \mathsf{hasChild.Human)}$ can be rewritten into the two smaller descriptions $\mathsf{Human} \sqcap \exists \mathsf{hasChild.Parent}$ and $\mathsf{Parent} \sqcap \exists \mathsf{hasChild.Parent}$, which are both equivalent to the original description.

Definition 4 (rewriting) *Let N_R be a set of role names and N_P a set of primitive names, and let $\mathcal{L}_s, \mathcal{L}_d$, and $\mathcal{L}_t$ be three DLs (the source-, destination, and TBox-DL, respectively). A rewriting problem is given by:*

- *an $\mathcal{L}_t$-TBox $\mathcal{T}$ containing only role names from N_R and primitive names from N_P; the set of defined names occurring in $\mathcal{T}$ is denoted by N_D,*
- *an $\mathcal{L}_s$-concept description C using only the names from N_R and N_P,*
- *a binary relation $\rho \subseteq \mathcal{L}_s \times \mathcal{L}_d$ between $\mathcal{L}_s$- and $\mathcal{L}_d$-concept descriptions.*

An $\mathcal{L}_d$-rewriting of C using $\mathcal{T}$ is an $\mathcal{L}_d$-concept description E built using names from N_R and $N_P \cup N_P$ such that $C \rho E$.

For example, consider that the three DLs are the language $\mathcal{ALN}$, and the relation ρ is instantiated by equivalence modulo $\mathcal{T}$. Let:

$$\begin{aligned}C &= \mathsf{Male} \sqcap \mathsf{Rich} \sqcap (\geq 1\ \mathsf{hasChild}) \sqcap \forall \mathsf{hasChild.(Male} \sqcap \mathsf{Rich)}\\ \mathcal{T} &= \{\ \mathsf{Father} \equiv \mathsf{Male} \sqcap \geq 1\ \mathsf{hasChild}),\\ &\quad \mathsf{RichParent} \equiv \mathsf{Rich} \sqcap \forall \mathsf{hasChild.Rich} \sqcap (\geq 1\ \mathsf{hasChild}),\\ &\quad \mathsf{FahterOfSons} \equiv \mathsf{Father} \sqcap \forall \mathsf{hasChild.Male}\ \}.\end{aligned}$$

Here, the concept description $\mathsf{FatherOfSons} \sqcap \mathsf{RichParent}$ is an $\mathcal{ALN}$-rewriting of C using $\mathcal{T}$.

3.2.2.5 Matching of Concept Descriptions

Matching is an inference service that allows to replace certain concept names by concept descriptions before testing for equivalence or subsumption [Küs01, Bra06].

Definition 5 (matching) *An $\mathcal{L}$-matching problem modulo equivalence and modulo subsumption is of the form $C \equiv^? D$ and $C \sqsubseteq^? D$ respectively, where C is a description and D a pattern. A solution or matcher of these problems is a substitution $\mathcal{A}$ such that $C \equiv \mathcal{A}(D)$ and $C \sqsubseteq \mathcal{A}(D)$, respectively.*

For instance, take the concept pattern $P : X \sqcap \exists \mathsf{hasChild}.X$, where X is a concept variable. Intuitively, this concept pattern speaks about people who share the same (unspecified) property X with one of their children. When X is substituted by Human, then P is equivalent to Parent. However, there is no substitution for X making P equivalent to Mother or Father, since the individuals of these concepts are required to be female or male respectively, whereas the children of a mother are not required to be female. Conversely, P matches against $\mathsf{Human} \sqcap \mathsf{Female} \sqcap \exists \mathsf{hasChild}.(\mathsf{Human} \sqcap \mathsf{Female})$.

3.2.2.6 Concept Contraction and Concept Abduction

Concept contraction [CNS+03] and concept abduction [NSDM03] are two inference services that were introduced in [CNS+05a] as a solution to find an optimal equilibrium between a demand (D) and a supply (S). The algorithm was implemented in a project for semantic-based discovery of matches and negotiation spaces in an e-marketplace [CNS+05c]. Also, principles for using this algorithm for a personalized E-Learning were published in [CNS+05b].

Concept contraction extends satisfiability. If the conjunction between the supply and the demand is unsatisfiable in the TBox $\mathcal{T}$, written $S \sqcap D \equiv_{\mathcal{T}} \bot$, then the aim is to retract requirements in D to obtain a concept K (for keep) such that $K \sqcap S \not\equiv_{\mathcal{T}} \bot$.

Definition 6 (concept contraction) *Let $\mathcal{L}$ be a DL, S, D be two concepts in $\mathcal{L}$, and $\mathcal{T}$ be a set of axioms in $\mathcal{L}$, where both S and D are satisfiable in $\mathcal{T}$. A concept contraction problem (CCP), identified by $< \mathcal{L}, S, D, \mathcal{T} >$ is finding a pair of concepts $< G, K > \in \mathcal{L} \times \mathcal{L}$ such that $D \equiv_{\mathcal{T}} G \sqcap K$, and $K \sqcap C$ is satisfiable in $\mathcal{T}$. K is called a contraction of D according to S and $\mathcal{T}$.*

Once contraction has been applied, and consistency between the supply and the demand has been regained, there is still the problem with partial specifications, i.e., it could be the case that the supply — though compatible — does not imply the demand. Then, it is necessary to assess what should be hypothesized in the supply in order to start the transaction with the demand. This non-standard inference is called *concept abduction* [NSDM03].

Definition 7 (concept abduction) *Let $\mathcal{L}$ be a DL, S, D be two concepts in $\mathcal{L}$, and $\mathcal{T}$ be a set of axioms in $\mathcal{L}$, where both S and D are satisfiable in $\mathcal{T}$. A concept abduction problem (CAP), identified by $< \mathcal{L}, S, D, \mathcal{T} >$, is finding a concept $H \in \mathcal{L}$ (hypotheses) such that $S \sqcap_{\mathcal{T}} H \sqsubseteq D$, and moreover $S \sqcap H$ is satisfiable in $\mathcal{T}$.*

Let us consider a simplified scenario in an e-marketplace. We have a demand (D) expressed as: "I am looking for a computer such that it must be a PC including an inkjet printer". We also have an available supply (S) expressed as: "Personal computer equipped with a high level laser printer". The aim is to find which parts are shared by D and S (K for keep), and which ones are not (G for give up). Formally, $D \equiv \mathsf{homePC} \sqcap \forall \mathsf{hasComponent}.\mathsf{InkjetPrinter}$, and $S \equiv \mathsf{homePC} \sqcap \forall \mathsf{hasComponent}.\mathsf{LaserPrinter}$. Solving a CCP we obtain a $< G, K >$, where $G = \forall \mathsf{hasComponent}.\mathsf{InkjetPrinter}$, and $K = \mathsf{homePC}$. Although S and D are not identical, $K \sqcap S$ is satisfiable, hence K potentially matches S.

3.2.2.7 Concept Cover

The concept covering problem [HLRT02] defines a cover of a concept description C w.r.t. a terminology $\mathcal{T}$ as being the conjunction of some defined concepts in $\mathcal{T}$ that share some information with a concept description Q (for query). Based on two non-standard inferences in DLs — i.e., the least common subsumer, and the semantic difference operation — a cover can be formally defined as follows:

Definition 8 (concept cover) *Let $\mathcal{L}$ be a DL with structural subsumption, $\mathcal{T}$ be an $\mathcal{L}$-terminology and $S_{\mathcal{T}} = \{S_i, i \in [1, n]\}$ the set of concept definitions occurring in $\mathcal{T}$. A cover of a $\mathcal{L}$-concept description $Q \not\equiv \bot$ using the terminology $\mathcal{T}$ is a conjunction E of some names S_i from $\mathcal{T}$ such that $Q - lcs(Q, E) \not\equiv Q$.*

For example, let $Q \equiv A_1 \sqcap A_2 \sqcap A_3 \sqcap A_4$ be a query over the following terminology:

$$\begin{aligned} C_1 &\equiv A_1 \sqcap A_2 \\ C_2 &\equiv A_2 \sqcap A_3 \\ C_3 &\equiv A_2 \sqcap A_3 \sqcap A_4 \\ C_4 &\equiv A_1 \end{aligned}$$

Then, a table of the concept covers can be drawn, which shows that the resulting best covers — even complete covers — are $C_3 \sqcap C_4$, and $C_1 \sqcap C_3$.

	A_1	A_2	A_3	A_4
C_1	×	×		
C_2		×	×	
C_3		×	×	×
C_4	×			

The algorithm has been implemented in the project MKBEEM (Multilingual Knowledge Based European Electronic Marketplace) [CGPL+03]. The concept covering problem is detailed in section 7.4.

3.2.3 Closed- vs. Open World Assumption

The *open world assumption* (OWA) assumes that its knowledge of the world is incomplete. If something cannot be proven to be true, then it does not automatically become false. In the OWA, what is not stated is considered unknown, rather than wrong. In contrary, the *closed world assumption* (CWA) is the presumption that what is not currently known to be true is false, see e.g., [GMP06]. For illustration, consider the statement "Serge is a citizen of Luxembourg", and the question: "Is Serge a citizen of Germany?". The CWA-answer is "no", but OWA-answer is "unkown".

SW languages such as RDF(S) and OWL implicitly make the OWA. In essence, from the absence of a statement alone a deductive reasoner cannot (and must not) infer that the statement is false [VVSH07].

3.3 Reasoning Algorithms

As pointed out in section 3.2, approaches for solving (non-)standard inference problems are usually based on satisfiability and subsumption. There are two types of algorithms for solving subsumption: *structural subsumption* and *tableau algorithms*. Structural subsumption algorithms compare the syntactic structure of concept descriptions. While they are usually very efficient, they are only complete for rather simple languages with little expressivity. In particular, DLs with (full) negation and disjunction cannot be handled by structural subsumption algorithms. For such languages, so-called tableau-based algorithms have turned out to be very useful.

3.3.1 Structural Subsumption

These algorithms usually proceed in two phases. First, the descriptions to be tested for subsumption are in a *normal form*. Secondly, the syntactic structure of the normal forms is compared. A concept description is in $\mathcal{EL}$-normal form iff it has the form:

$$A_1 \sqcap ... \sqcap A_n \sqcap \ \exists R_1.C_1 \sqcap \ \exists R_m.C_m,$$

where $A_1 \sqcap ... \sqcap A_n$ are distinct concept names, $R_1 \sqcap ... \sqcap R_m$ are distinct role names, and $C_1 \sqcap ... \sqcap C_m$ are concept descriptions in normal form. In other words, a concept description is in its $\mathcal{EL}$-normal form if it does not contain any redundant information.

Definition 9 (structural subsumption) *Let C, D be two concept descriptions in their normal form, i.e.,*

$$C \equiv A_1 \sqcap ... \sqcap A_m \sqcap \ \exists R_1.C_1 \sqcap \ \exists R_n.C_n,$$

$$D \equiv B_1 \sqcap ... \sqcap B_k \sqcap \ \exists S_1.D_1 \sqcap \ \exists S_l.D_l,$$

then $C \sqsubseteq D$ iff the following two conditions hold:

- $\forall 1 \leq i \leq k, \exists 1 \leq j \leq m : B_i = Aj,$
- $\forall 1 \leq i \leq l, \exists 1 \leq j \leq n : S_i = R_j$ *and* $C_j \sqsubseteq D_i$.

Consider the following concept descriptions, where $C \sqsubseteq D$ with structural subsumption holds:

$$C \equiv \mathsf{Human} \sqcap \mathsf{Female} \sqcap \mathsf{Woman} \sqcap \ \exists \mathsf{hasChild}.\mathsf{Woman},$$

$$D \equiv \mathsf{Human} \sqcap \mathsf{Woman} \sqcap \ \exists \mathsf{hasChild}.\mathsf{Human}.$$

A classical example where structural subsumption fails is the following:

$$A \sqcap \neg A \sqsubseteq^? B \sqcap \neg B,$$

which results in testing:

$$(A \sqsubseteq^? B \vee A \sqsubseteq^? \neg B) \wedge (\neg A \sqsubseteq^? B \vee \neg A \sqsubseteq^? \neg B).$$

The result would be "no"; none of the four subsumption tests holds. However, $A \sqcap \neg A \equiv \bot$ and $B \sqcap \neg B \equiv \bot$. Therefore, $A \sqcap \neg A \sqsubseteq B \sqcap \neg B$ results in testing if $\bot \sqsubseteq \bot$ holds, which is the case.

3.3.2 Tableau Algorithms

Tableau algorithms are usually employed for DLs that allow full negation. Testing subsumption is reduced to deciding satisfiability of concepts. As stated in section 3.2.1.2, subsumption can always be transformed in testing satisfiability for DLs with full negation. For instance, $C \sqsubseteq D$ iff $C \sqcap \neg D \equiv \bot$.

We will not elaborate on this subject because our solution depicted in chapter 7 is based on algorithms with structural subsumption, and refer the interested reader to [BCM+03].

3.4 OWL and Description Logics

OWL is based in part on DLs (see section 2.2.4), i.e., $\mathcal{SHOIN}$(D) — where D stands for a datatype theory — and also on a number of earlier knowledge representing systems known as frame-based systems [HPS04, CNS+05a, MSS05]. Its subset OWL Lite is based on the less expressive logic $\mathcal{SHIF}(D)$. All reasoning tasks in both OWL DL and OWL Lite can be reduced to knowledge based satisfiability. OWL Full operates outside the bounds of DLs, allowing more power and expressivity and having fewer constraints on use, but at the cost of decidability.

The OWL DL-sublanguages are defined as follows:

$$
\begin{array}{lcl}
\mathcal{S} & \rightarrow & \mathcal{ALC} \ | \ R_{+}(\text{transitive roles}) \\
\mathcal{H} & \rightarrow & \text{for role hierarchy, e.g., } \mathsf{hasDaughter} \sqsubseteq \mathsf{hasChild} \\
\mathcal{O} & \rightarrow & \text{for nominals/singleton classes, e.g., } \{\mathsf{Italy}\} \\
\mathcal{I} & \rightarrow & \text{for inverse roles, e.g., } \mathsf{isChildOf} \equiv \mathsf{hasChild}^{-} \\
\mathcal{N} & \rightarrow & \text{for number restrictions, e.g., } \geq 2\mathsf{hasChild}, \leq 2\mathsf{hasChild} \\
\mathcal{Q} & \rightarrow & \text{for qualified number restrictions, e.g., } \geq 2\mathsf{hasChild.Doctor}
\end{array}
$$

In DLs, a datatype theory D is a mapping from a set of datatypes to a set of values, e.g., from `xsd:integer` to the integers, plus a mapping from data values to their denotation, which must be one of the set of values, e.g., from `"1"^^xsd:integer` to the integer 1. The datatype (or concrete) domain, written $\Delta_{D}^{\mathcal{I}}$, is the union of the mappings of the datatypes.

DL can be transformed in a machine readable form, i.e., OWL without losing any of its details. For example, consider the DL-concept description shown in (3.1). Its serialization as an OWL-XML file is shown in figure 3.6.

```
<owl:Class rdf:about="#Mother">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:resource="#Woman" />
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasChild" />
      <owl:someValuesFrom rdf:resource="#Human" />
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Figure 3.6: Example of an OWL serialization.

Chapter 4

Search Engines and QA-Systems

Contents

The search engine allows one to ask for content meeting specific criteria — typically by using keywords — and retrieves a list of items that match those criteria. This list is often sorted w.r.t. some means of relevance. Question-answering is regarded as requiring more complex natural language processing techniques than other types of information retrieval systems, and it is sometimes regarded as the next step beyond search engines.

This chapter will provide an overview of search engines in section 4.1, and question-answering system in section 4.2. Some state of the art search engines and question-answering systems will be described in section 4.3. A more complete study can be found in [MHG$^+$02, Lew05].

4.1 Search Engines

A search engine is an information retrieval system designed to help find information stored on a computer system, such as on the WWW, inside a corporate or proprietary network, or in a personal computer. The search engine allows one to ask for content meeting specific criteria — typically by using keywords — and retrieves a list of items that match those criteria. This list is often sorted w.r.t. some means of relevance.

4.1.1 History

The very first tool used for searching on the Internet was *Archie*[1]. The name stands for "archive" without the "v". It was created in 1990 by Alan Emtage, a student at McGill University in Montreal. The program downloaded the directory listings of all the files located on public anonymous FTP-sites, creating a searchable database of filenames. Archie could not search by file contents.

The first Web search engine was *Wandex*, a now-defunct index collected by the WWW Wanderer, a Web crawler developed by Matthew Gray at MIT in 1993. Another very early search engine, *Aliweb*[2] (Archie Like Indexing for the Web), also appeared in 1993, and still runs today.

Soon after, many **first generation** search engines (1995 – 1997) appeared and vied for popularity. These tools were using almost only on-page data such as text and formatting information to compute result ranking. The **second generation** of search engines (since 1998) are using off-page, Web-related data such as link analysis, anchor-texts, and click-through data, e.g., Google. The **third generation** of search engines (since 2003) try to blend data from multiple, heterogeneous sources trying to answer "the need behind the query" [Bro02]. The computed results are customized according to the user's information needs, taking into account the user's personal data background, context, and intention. They include social networking information, tagging, user feedback, semantic analysis, recommendations, and trustworthiness of information (according to its source).

4.1.2 How search engines work

Web search engines work by storing information about a large number of Web pages. These pages are retrieved by a "Web crawler" — sometimes also called spider, robot, or agent — that is an automated Web browser which follows every link it sees. The contents of each page are then analyzed to determine how it should be indexed, e.g., words are extracted from the titles, headings, or meta tags.

Data about Web pages are stored in an index database for use in later queries. Some search engines, such as Google — which is currently the most popular search engine on the Web according to Nielsen NetRatings[3] — store all or part of the source page, as well as information about the Web pages; it is commonly called "Google cache". Other search engines such as AltaVista store every word of every page they find.

When a user enters a query into a search engine (typically by using keywords), the engine examines its index, and provides a listing of best-matching Web pages, usually with a short summary containing the document's title and sometimes parts of the text. Most search engines support the use of Boolean operators AND, OR, and NOT to further specify the search query. Some search engines provide an advanced feature called *proximity search*, which allows users to define

[1] `http://archie.icm.edu.pl/`
[2] `http://www.aliweb.com/`
[3] `http://www.nielsen-netratings.com/pr/pr_070620.pdf`

the distance between keywords. For instance, a search could be used to find "red brick house", and match phrases such as "red house of brick" or "house made of red brick". By limiting the proximity, these phrases can be matched while avoiding documents where the words are scattered, or spread across a page or in unrelated articles in an anthology.

The usefulness of a search engine depends on the relevance of the result set it gives back. While there may be millions of Web pages that include a particular word or phrase, some pages may be more relevant, popular, or authoritative than others. Most search engines employ methods to rank the results to provide the "best" results first. How a search engine decides which pages are the best matches, and what order the results should be shown in, varies widely from one engine to another. One of the most popular techniques is *PageRank* by Google. It is described as follows[4]:

> PageRank relies on the uniquely democratic nature of the Web by using its vast link structure as an indicator of an individual page's value. In essence, Google interprets a link from page A to page B as a vote, by page A, for page B. But, Google looks at more than the sheer volume of votes, or links a page receives; it also analyzes the page that casts the vote. Votes cast by pages that are themselves "important" weigh more heavily and help to make other pages "important".

4.1.3 Problems with Current Search Engines

Search engines on the Web are faced by important challenges. This is an incomplete list that was found in the Wikipedia[5]:

- The Web is growing much faster than any present search engine can possibly index.
- Many Web pages are updated frequently, which forces the search engine to revisit them periodically.
- Most search engines limit their queries to keywords, which may result in many false positives, especially using the default whole-page search.
- Dynamically generated sites may be slow or difficult to index, or may result in excessive results, perhaps generating 500 times more Web pages than average.
- Many dynamically generated Web sites are not indexable by search engines; this phenomenon is known as the *invisible Web*[6]
- Relevancy: sometimes the search engine cannot get what the person is looking for.
- Some search engines do not rank results by relevance, but by the amount of money the matching Web sites pay.
- In 2006, hundreds of generated Web sites used tricks to manipulate a search engine to display them in the higher results for numerous keywords.
- Secure pages (content hosted on HTTPS URLs) pose a challenge for crawlers, which either cannot browse the content for technical reasons, or will not index it for privacy reasons.

[4] `http://www.google.com/technology/`

[5] `http://en.wikipedia.org/wiki/Search_engine#Challenges_faced_by_search_engines`

[6] `http://en.wikipedia.org/wiki/Deep_web`

4.2 Question-Answering Systems

Question-answering (QA) is a type of information retrieval. Given a collection of documents such as the WWW or a local collection, the system should be able to retrieve answers to questions posed in natural language (NL). QA is regarded as requiring more complex NL processing techniques than other types of information retrieval such as document retrieval, and it is sometimes regarded as the next step beyond search engines.

Closed-domain QA deals with questions under a specific domain, e.g., computer history, fractions in mathematics, or networks in computer science, and can be seen as an easier task, because NL processing systems can exploit domain-specific knowledge frequently formalized in ontologies. *Open-domain* QA deals with questions about nearly everything and can only rely on general ontologies and world knowledge (see section 2.3.3). On the one hand, the computation of pertinent and reliable answers is much more complex than in closed-domain QA, but on the other hand, these systems usually have much more data available from which to extract the answer.

4.2.1 History

Some of the early artificial intelligence systems were QA systems. A good overview of early QA systems is given in [HG01]. Two of the most famous QA systems of that time are *Baseball* and *Lunar*, both of which were developed in the 1960s. Baseball answered questions about the US baseball league over a period of one year. Lunar, in turn, answered questions about the geological analysis of rocks returned by the Apollo moon missions. Both QA systems were very effective in their chosen domains. In fact, Lunar was demonstrated at a lunar science convention in 1971, and it was able to answer 90% of the questions in its domain posed by people untrained on the system.

The 1970s and 1980s saw the development of comprehensive theories in computational linguistics, which led to the development of ambitious projects in text comprehension and QA. One example of such a system was the *Unix Consultant* [WAC84], a system that answered questions pertaining to the Unix operating system. The system had a comprehensive hand-crafted knowledge base of its domain, and it aimed at phrasing the answer to accommodate various types of users. The system developed in the Unix Consultant project never went past the stage of simple demonstrations, but it helped the development of theories on computational linguistics and reasoning.

4.2.2 How Question-Answering Systems Work

QA is very dependent on a good search corpus; for without documents containing the answer, there is little any QA system can do. It thus makes sense that larger collection sizes generally lend well to better QA performance, unless the question domain is orthogonal to the collection.

Some methods of QA use keyword-based techniques to locate interesting passages and sentences from the retrieved documents, and then filter them according to the presence of the desired answer type within that candidate document. Ranking is then done with regard to syntactic features such as word order or location, and similarity to query.

When using massive collections with good data redundancy, some systems use templates to find the final answer in the hope that the answer is just a reformulation of the question. If you posed the question “Who invented the transistor?”, the system would detect the substring “Who invented the X”, and look for documents which start with “X invented the Y”. This often works well on simple questions seeking factual tidbits of information such as names, dates, locations, and quantities. We adopted this solution in what we will describe as “NLP strategy 2” in our E-Librarian Service (see section 7.3).

However, in the cases where simple question reformulation or keyword techniques will not suffice, more sophisticated syntactic, semantic, and contextual processing must be performed to extract or construct the answer. These techniques might include named-entity recognition, relation detection, coreference resolution, syntactic alternations, word sense disambiguation, logic form transformation, logical inferences, temporal or spatial reasoning, etc. These systems will also very often utilize ontologies to augment the available reasoning resources through semantic connections and definitions. We adopted such techniques in what we will describe as "NLP strategy 3" in our E-Librarian Service (see section 7.3).

4.3 Some State of the Art Systems

In this section, we will present some state of the art search engines and QA systems that can also be considered as related work.

4.3.1 Cyc

One of the first interactive knowledge bases is *Cyc*[7]. The project was started in 1984 by Doug Lenat as part of Microelectronics and Computer Technology Corporation. The name "Cyc" comes from "encyclopedia". Cyc is a formalized representation of a vast quantity of fundamental human knowledge: facts, rules of thumb, and heuristics for reasoning about the objects and events of everyday life. Cyc can be queried in NL.

The original knowledge base is proprietary, but a smaller version — intended to establish a common vocabulary for automatic reasoning — was released as *OpenCyc*[8] under an open source license. More recently, Cyc has been made available to researchers under a research-purposes license as ResearchCyc[9].

General knowledge bases like Cyc figured out to have the same major disadvantage: they contain too much general information, and often lack specific domain knowledge. There is a large number of gaps in not only the ontology of ordinary objects, but an almost complete lack of relevant assertions describing such objects. Also, the system is very complex which results in scalability problems, e.g., it is very difficult to add new items to the knowledge base manually.

4.3.2 START

START[10] [Kat97] is the first QA system available on the WWW. It has been developed by Boris Katz and his associates of the InfoLab Group at the MIT Computer Science and Artificial Intelligence Laboratory. Several improvements have been made since it came online in 1993, e.g., [KL02, KFY+02].

Currently, the system can answer millions of English questions about places (e.g., cities, countries, lakes, coordinates, weather, maps, demographics, political and economic systems), movies (e.g., titles, actors, directors), people (e.g., birth dates, biographies), dictionary definitions, etc.

Although START can be queried in NL, it seems that no advanced computations are performed to extract the semantic meaning of the query. For example, the question "Who invented the transistor?" yields two answers: the inventors of the transistor, but also a description of the transistor (the answer to the question: "What is a transistor").

[7] `http://www.cyc.com/`
[8] `http://www.opencyc.org/`
[9] `http://research.cyc.com/`
[10] `http://start.csail.mit.edu/`

4.3.3 AquaLog

AquaLog[11] [LPM05] is a portable question-answering system which takes queries expressed in NL and an ontology as input, and returns answers drawn from one or more knowledge bases. User questions are expressed as triples: <subject, predicate, object>. If the several translation mechanisms fail, then the user is asked for disambiguation. The system also uses an interesting learning component to adapt to the user's "jargon".

AquaLog has currently a very limited knowledge space. In a benchmark test over 76 different questions, 37 (48.68%) were handled correctly.

4.3.4 Precise

The prototype PRECISE [PEK03] uses ontology technologies to map semantically tractable NL questions to the corresponding SQL query. It was tested using several hundred questions drawn from user studies over three benchmark databases. Over 80% of the questions are semantically tractable questions which PRECISE answered correctly, and it recognized the 20% it could not handle and requested a paraphrase.

The problem of finding a mapping from the tokenization to the database requires that all tokens must be distinct; questions with unknown words are not semantically tractable and cannot be handled.

4.3.5 Falcon

FALCON [WFSP00] is an answer engine that handles questions in NL. When the question concept indicating the answer type is identified, it is mapped onto an answer taxonomy. The top categories are connected to several word classes from WordNet[12]. FALCON gives a cached answer if a similar question has already been asked before; a similarity measure is calculated to see if the given question is a reformulation of a previous one. User feedback is also incorporated in the system.

In TREC-9, FALCON generated a score of 58% for short answers and 76% for long answers, which was actually the best score.

4.3.6 Ask.com

Ask.com[13] (see figure 4.1) was originally known as *Ask Jeeves*, where "Jeeves" is the name of the "gentleman's gentleman" fetching answers to any question asked. The original idea behind Ask Jeeves was to allow users to get answers to questions posed in everyday NL. Ask.com was the first commercial search engine for the WWW. It supports a variety of user queries in plain English, as well as traditional keyword searching, and strives to be more intuitive and user-friendly than other search engines. The *ExpertRank* algorithm provides search results ordered through attempting to identify authoritative Web sites. Furthermore, link popularity, and subject-specific popularity are also considered. Topics are identified using experts on those topics. This information is used to help improve the ordering of returned Web sites during searches.

Like most of the Web search engines, Ask.com manages ambiguities badly. For instance, the question "Who invented Ada?" returns nearly 40,000 answers, while most of them are not related to computer science. The task of filtering the pertinent information out of the *noise* still remains an awkward user-task.

[11] `http://kmi.open.ac.uk/technologies/aqualog/`
[12] WordNet is described in section 6.4.1.
[13] `http://www.ask.com`

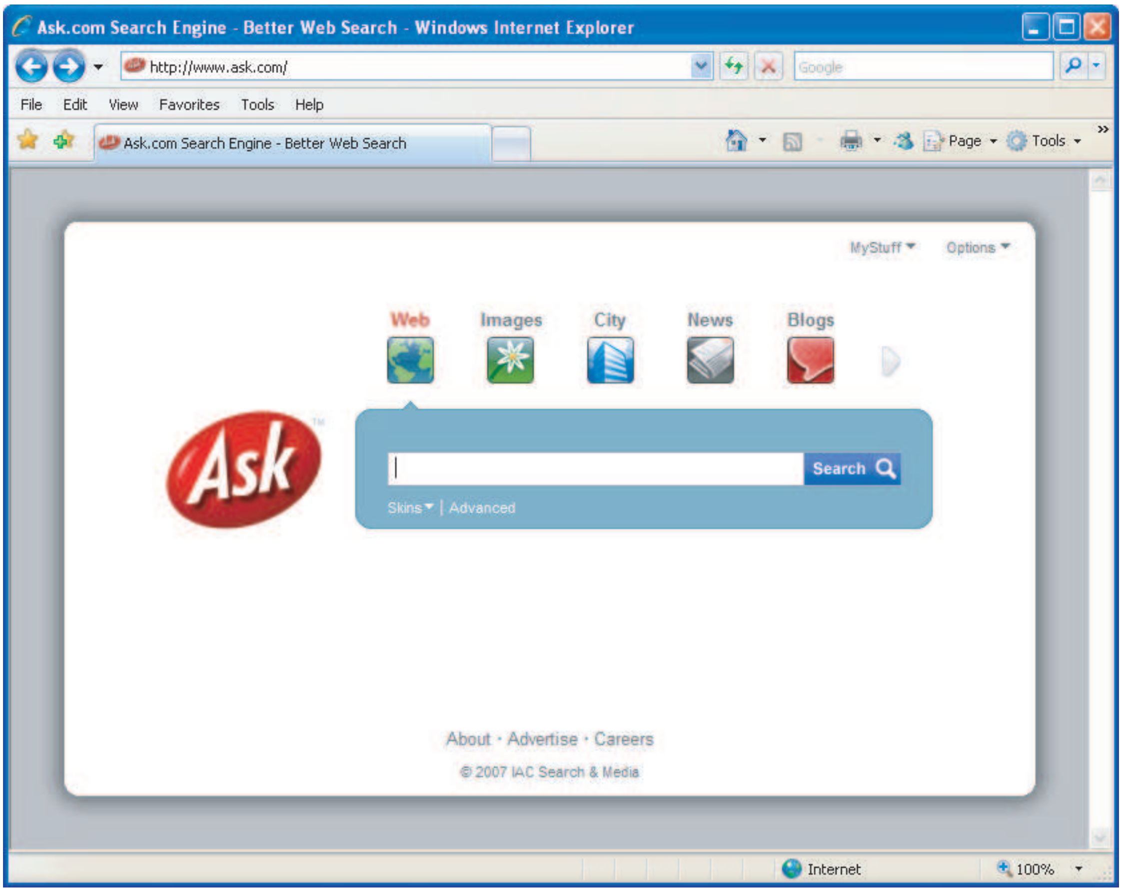


Figure 4.1: Web interface of the search engine Ask.com.

4.3.7 askEd!

askEd![14] is the prototype of an on-going research project by Ed Whittaker in the Furui Laboratory at Tokyo Institute of Technology. A purely statistical, data-driven, and non-linguistic approach to the problem of QA was adopted in askEd!. Instead, a large amount of data is used in the hope that somewhere there is some text in a form that more-or-less matches the question, and allows to extract the answer. This simple approach was implemented in the prototype for different languages: English, Russian, Japanese, and Chinese.

Currently, the knowledge base is very limited. The prototypical system can only answer a short set of simple questions.

4.3.8 searchCrystal

searchCrystal[15] is a promising project that searches and compares multiple engines in one place. It is a search visualization tool that compares the best Web, image, video, blog, tagging, news engines, or RSS feeds. Other features of searchCrystal are that it can be embedded on a Web site or blog to share personalized crystals, and it can be used to find out what is popular on Wikipedia. Currently, searchCrystal is only available in a demo version.

[14] `http://asked.jp/`
[15] `http://www.searchcrystal.com/`

4.3.9 Other Search Engines and QA Systems

A lot of other search engines and QA systems exist like: *PowerAnswer*[16] [MHG+02], *AnswerBus*[17] [Zhe02], *BrainBoost*[18], *Factoid*[19] [RRF04], *TellMe*[20] [PM05], *Geoquery!*[21], *KnowItAll*[22] [ECD+04], *PowerSet*[23], *TextRunner*[24] [BCS+07], *Querix*[25] [KBZ06], *ActiveMath*[26] [MS04b], *Hakia*[27], Osotis[28] [SW06b], and *Mulder* [KEW01].

Two complete systems for recording, annotating, and retrieving multimedia documents are *LectureLounge* and *MOM*. LectureLounge [WPS+04] is a system to automatically and non-invasively capture, analyze, annotate, index, archive, and publish live-presentations. MOM (Multimedia Ontology Manager) [BBT+06] is a system that allows the creation of multimedia ontologies, supports automatic annotation and creation of extended commentaries of video sequences, and permits complex queries by reasoning over the ontology.

[16] `http://www.languagecomputer.com/`
[17] `http://www.answerbus.com/`
[18] `http://www.brainboost.com/`
[19] `http://qa.wpcarey.asu.edu/`
[20] `http://www.ics.mq.edu.au/~pizzato/tellme`
[21] `http://www.cs.utexas.edu/users/ml/geo.html`
[22] `http://www.cs.washington.edu/research/knowitall/`
[23] `http://www.powerset.com/`
[24] `http://www.cs.washington.edu/research/textrunner/`
[25] `http://www.ifi.uzh.ch/ddis/research/semweb/talking-to-the-semantic-web/querix/`
[26] `http://www.activemath.org/`
[27] `http://www.hakia.com/`
[28] `http://www.osotis.de/`

Part II

E-Librarian Service

Chapter 5

Ontological Approach for our E-Librarian Service

Contents

It has been realized that digital libraries could benefit from having its content understandable and available in a machine processable form, and it is widely agreed that ontologies will play a key role in providing much enabling infrastructure to achieve this goal [SS04, BCT07, AvH04, Fen04].

In this chapter, we will provide an overview of the ontological approach for our E-Librarian Service. Section 5.1 will give a brief overview of ontology driven systems. We will describe the general ontological approach of our E-Librarian Service in section 5.2, and the semantic annotation of the knowledge base in section 5.3.

5.1 Ontology Driven Systems

This section will give a general overview of ontology driven systems, while the foundations of ontologies have already been provided in section 2.3.

Users rather accept the results of a computer tool if it is able to explain its reasoning [CEE+01]; such systems are commonly referred to as *expert systems*. An expert system is a system that employs knowledge about its application domain, and uses an inferencing (reasoning) procedure to solve problems that would otherwise require human competence or expertise. Such specific systems rely on a specialized and hierarchically organized knowledge base, and a specific reasoning engine. Question-answering systems can be a kind of expert systems. In the recent years it has become apparent that ontologies will play a key role in such systems, particularly in the field of knowledge management [Fen04]. Ontologies enable effective and efficient access to heterogeneous and distributed knowledge bases. Here are some examples:

- The prototype PRECISE (see section 4.3.4) uses ontology technology to map semantically tractable natural language questions to the corresponding SQL query.
- An ontology-driven semantic search is presented in [BCFB04] that allows to set up semantic level relevance feedback for query concept focalization, generalization, etc.
- The results of an ontology-based question-answering system in chemistry called *OntoNova* are reported in [AMO+03]. The system is able to logically infer over the domain specific knowledge base, and to justify its answers giving natural language explanations.
- A domain ontology information retrieval system based on speech recognition is presented in [TGF+07].
- A system for reasoning over multimedia E-Learning objects is described in [EHLS06]. Here, a speech recognition engine is used for keyword spotting. It extracts the taxonomy node that corresponds to the keyword, and associates it to the multimedia objects as metadata.

5.2 Ontological Approach

A fundamental part of our E-Librarian Service is a domain ontology. It is used:

- for the semantic annotation of the documents in the knowledge base (see section 5.3),
- for the translation of the users' questions from natural language into a logical form (see section 6.4),
- for the retrieval of semantically pertinent documents from the knowledge base (see section 7.4).

An existing ontology can be used, or one can build its own ontology that is optimized for the used knowledge base(s). An overview of ontological engineering is given in [Gru95, GPCGFL03, RDH+04, HV05, ÖS05].

Different prototypes were developed during this research work (see chapter 9); one about computer history, one about fractions in mathematics, and one about networks in computer science. As far as we know, no ontology about these precise domains exists. In order to contribute to current ontology research, we created three ontologies for these domains; two rather small ones

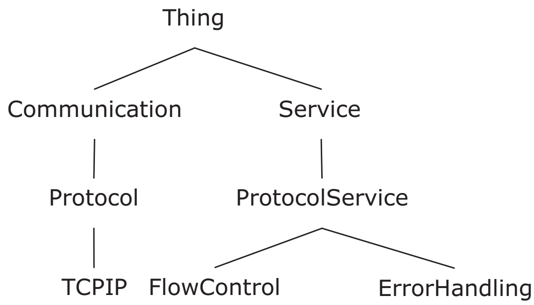


Figure 5.1: Sample of a taxonomy about networks in computer science.

about computer history and fractions in mathematics, and one relatively large one about networks in computer science.

For the remaining part of this thesis, we will always refer to the prototype and ontology about computer networks. Figure 5.1 illustrates a part of the taxonomy. Here, a document describing the protocol TCP/IP would be placed in the concept TCPIP, which is a specialization of the concept Protocol.

On the one hand, the more detailed the taxonomy is, the more exact the system can classify the documents. On the other hand, a very detailed taxonomy reduces the tolerance of the NL processing; the users' questions must be very well and precisely formulated (see section 6.4).

Our ontology was created with Protégé[1] [KFNM04, KMR04]; an illustration is shown in figure 5.2. Currently, the ontology contains 608 concepts and 26 roles.

5.3 Knowledge base annotation

An ontology has a well-structured knowledge base over which inference is possible. Therefore, the expert systems are able to find implicit consequences of its explicitly represented knowledge. A constraint of such systems is that the content of the knowledge base must be semantically described by additional data — called metadata — in a machine readable form; we used OWL. OWL was described in section 2.2.

This project focuses on the elaboration and the study of an E-Librarian Service, not on the automatic extraction or generation of metadata (the latter is briefly covered in section 15.2). Therefore, we manually created an optimal annotation for the knowledge base w.r.t. a given domain ontology. However, to simplify this task, we developed a tool to assist the person that creates the metadata; we call him/her "the administrator". Here is a brief description of how the tool works.

- A list of all words that are used in the documents is automatically generated; they are extracted from the Powerpoint files. We developed a tool[2] to convert Powerpoint files into pure text. Only the words from the slides are considered, not the ones in the audio transcript. The extracted words are then transformed into their lemmatized (canonical) form, and doubles are deleted.

- The administrator classifies the lemmatized word into the ontology. Words that are not

[1] `http://protege.stanford.edu/`
[2] `http://www.linckels.lu/logiciels/ppt2txt.zip`

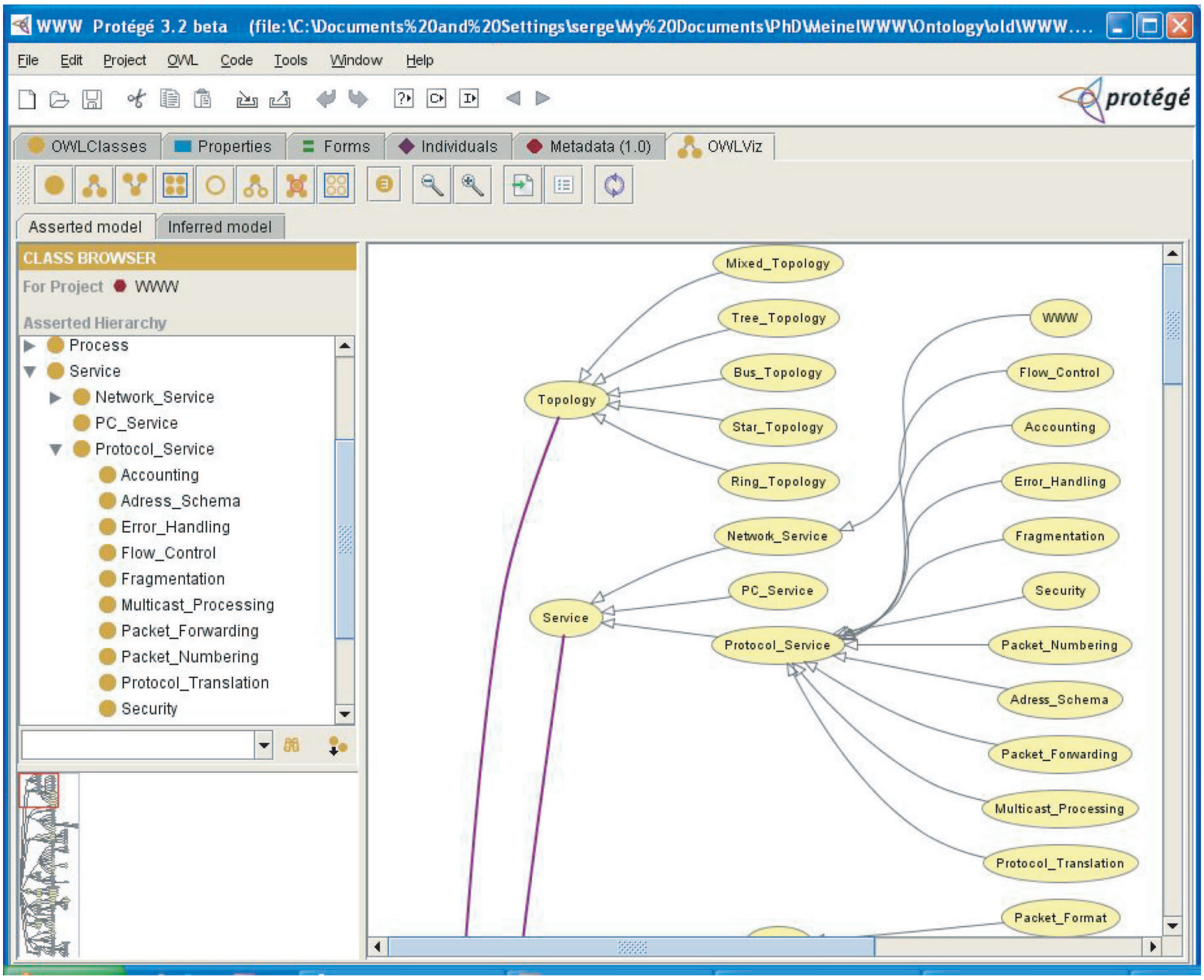


Figure 5.2: The tool Protégé was used to create the ontology about networks in computer science.

relevant for the given domain are ignored; we call them "stop words". The result of this classification is what we will refer to as "domain dictionary" (see section 6.4.1).

- The semantic annotation for each document is created by providing pertinent ontology concepts and roles. In other words, for each document the administrator formulates a DL-concept description that best describes the content of the document. We created a tool which visualizes the available ontology concepts and roles, and assists the administrator to build a valid expression. Finally, this DL-concept description is serialized as an OWL file (an example is shown in figure 3.6).

Chapter 6

Natural Language Processing

Contents

Natural langauge processing studies the problems inherent in the processing and manipulation of natural language and natural langauge understanding devoted to making computers "understand" statements written in human languages. One important and required feature of our E-Librarian Service is to allow users to enter complete questions; to simplify the human-machine communication, and to enable the search engine to better "understand" the users' queries.

The processing of a user query in natural language will be described in this chapter. We will start in section 6.1 with a short overview of natural language processing in computer science. Then we will summarize our contributions in section 6.2. Four different strategies that we explored to process the users' questions will be presented in section 6.3. Section 6.4 will detail the retained strategy, and present our algorithm for the translation of a complete user question into a computer readable and non-ambiguous form.

6.1 Natural Language Processing in Computer Science

> Natural language processing (NLP) is a subfield of artificial intelligence (AI) and linguistics. It studies the problems inherent in the processing and manipulation of natural language (NL), and NL understanding devoted to making computers "understand" statements written in human languages.

One goal of AI work in NL is to enable communication between people and computers in a natural "human-like" way, e.g., by means of verbal communication. NL understanding is sometimes referred to as an AI-complete problem[1], because NL recognition seems to require extensive knowledge about the outside world, and the ability to manipulate it. The definition of "understanding" is one of the major problems in NLP.

When we as humans process language, we are continually making guesses about meaning, using our rich knowledge of the world and of the current culture to try and work out what is being communicated. For example, if asked "Is there water in the fridge?", most humans would consider this question as referring to a bottle of mineral water in the fridge. In fact, we use language with a rich knowledge of "normal circumstances". When using language in these situations, we do not need to state them explicitly. They are assumed to be known to all communicators. This unspoken context may comprise 90% of a communication, and allows the language which builds on this foundation to be very concise [Inm04]. But computers cannot hold even one reasonably rich view of context, culture, or normal circumstances. For a computer, everything is an endless muddle of possibility with almost no way to sort out the "normal circumstance".

The importance and the benefits of NLP — especially for the Semantic Web and ontologies — is a topic that is only poorly covered in literature. On the one hand, it seems that NL interfaces to applications have become more acute with new technologies, e.g., from the Semantic Web and computational linguistics [CRFRJFN05, ART95, Pop05, WLZR07, CNS+05c, CE05]; nontechnical people could access information through their Web browsers, PDAs, cell phones, navigation systems, etc. in a very easy and user-friendly way. On the other hand, following several people's point of view, improving the annotation and the representation of the knowledge is more promising than NLP[2].

6.2 Objective and Contributions

One of the objectives of this project is to investigate in how far a search engine would yield more pertinent results if the query was a complete question in NL instead of keywords only. Our motivation is twofold.

- Firstly, most people that are not search experts have difficulties or are not able to formulate their query in a machine optimized way, e.g., by combining search terms with Boolean operators. It is also possible that they do not use the right domain expressions. A NL interface would simplify the human-machine interaction, which is especially useful in an educational environment (see chapter 11). Our E-Librarian Service allows the user to freely formulate a question in NL. This enables users to focus on *what* they want, rather than to worry about *how* and *where* to obtain the answer *from*.

[1] AI-complete is, by analogy to NP-completeness in complexity theory to indicate that the difficulty of a computational problem is equivalent to solving the central AI problem, i.e., making computers as intelligent as people.

[2] This statement results from several personal discussions.

- Secondly, in order to create an "intelligent" search mechanism, the user must enter a query which contains enough semantics, so that the E-Librarian Service understands the sense of the question in a non-ambiguous way, and is able to logically infer over the knowledge base. In principle, a complete question in NL contains more semantics than just keywords. Our E-Librarian Service considers linguistic information within the user question and the given context from the domain ontology to understand the sense of the sentence, and to translate it into a logical form.

6.3 Explored Strategies

In this section, we will give an overview of four NLP strategies that we explored. All strategies have in common that they use a domain ontology, and transform the NL user question into a logical and computer-readable form.

6.3.1 Strategy 1

Each concept in the ontology refers to a set of semantically relevant words. The idea is to map each word (token) in the user question to one or more concepts in the ontology. Semantically unimportant words are not mapped to a concept, thus will not be considered in the latter retrieval. The logical form of the user question is then defined by the conjunction of the mapped concepts and their respective values in the original sentence. For example, the question "Wer hat den Transistor erfunden?" (Who invented the transistor?) would be processed like this:

Token		Ontology concept
Wer	→	concept: Person
hat	→	∅
den	→	∅
Transistor	→	instance of concept: ECompoment
erfunden	→	role: wasInventedBy

This strategy was implemented in CHESt v2 (see chapter 9), published in [LM04a, LM04b, LM05c], and tested in experiments (see chapter 11). It turned out that this straightforward solution returns reliable results for simple and precise questions, but not for more complex and general questions.

6.3.2 Strategy 2

The simple mapping of isolated words to concepts is improved by considering linguistic information in the query as is done in [MBR01]. The idea is to read the *word categories* for each token in the user question, e.g., verb, noun, or article. It is then possible to map categories to taxonomy concepts, and to ignore semantically irrelevant word categories (e.g., article). The linguistic pre-processing is performed with a *part-of-speech (POS) tagger.* Most POS taggers are based on statistical observations, and trained over large corpora.

We tested the following taggers and NLP tools: Alembic Workbench[3], Apple Pie Parser[4],

[3] `http://www.mitre.org/tech/alembic-workbench/`
[4] `http://nlp.cs.nyu.edu/app/`

Babel[5], Brill Tagger[6], SCOL[7], CtxMatch[8], Fastr[9], Gate[10], Java WordNet Library (JWNL)[11], Connexor Machinese[12], MXPOSR Tagger[13], Heart of Gold[14], QTag[15], Sleepy Student Parser[16], Stanford Lexicalized Parser[17] Trainable Information Extractor (TIE)[18], TiMBL Tagger[19], Trigrams'n'Tags (TnT)[20], TreeTagger[21], Xerox XRCE Tagger[22]. We retained *TreeTagger* because it is a lightweight, reliable, and easy to use tagger. The authors[23] also granted us permission to modify the dictionary so that we could adapt it to our needs.

The following example illustrates how this NLP strategy works: the question "Wer hat den Transistor erfunden?" (Who invented the transistor?) would be processed like this:

Token		Lemma	Word category	Taxonomy concept
Wer	→	wer	PWS	concept: Person
hat	→	haben	VAFIN	∅
den	→	den	ART	∅
Transistor	→	Transistor	NN	instance of concept: EComponent
erfunden	→	erfinden	VVPP	role: wasInventedBy

This strategy was implemented in CHESt v3 and in MatES (see chapter 9), published in [LM06b, LM06c, LM07], tested in benchmark tests (see chapter 8), and used in an experiment (see chapter 12).

6.3.3 Strategy 3

Motivated by the hypothesis that a more advanced analysis of the user question might result in better search results, a new NLP strategy was explored. The idea was to consider the *syntax* of a sentence, and to read linguistic relations between the words in the sentence as it is done in [TM01, KEW01].

This strategy improved the mapping algorithm considerably. In fact, the syntactic structure of a sentence indicates the way how words in the sentence are related to each other, e.g., how the words are grouped together into phrases, which words modify which other words, and which words are of central importance in the sentence. Most syntactic representations of language are based on the notion of *context-free grammars* (CFG) [Sch03], which represent the sentence structure in

[5] http://www.cl.uni-bremen.de/~stefan/Babel/
[6] http://www.cs.jhu.edu/~brill/
[7] http://www.research.att.com/~abney/
[8] http://dit.unitn.it/~zanobini/
[9] http://www.limsi.fr/Individu/jacquemi/FASTR/
[10] http://gate.ac.uk/
[11] http://sourceforge.net/projects/jwordnet
[12] http://www.connexor.com/
[13] http://www.cis.upenn.edu/~adwait/statnlp.html
[14] http://heartofgold.dfki.de/
[15] http://www.english.bham.ac.uk/staff/omason/software/qtag.html
[16] http://www.coli.uni-saarland.de/~adubey/sleepy/
[17] http://nlp.stanford.edu/downloads/lex-parser.shtml
[18] http://www.inf.fu-berlin.de/inst/ag-db/software/ties/
[19] http://ilk.uvt.nl/timbl/
[20] http://www.coli.uni-saarland.de/~thorsten/tnt/
[21] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/
[22] http://www.xrce.xerox.com/competencies/content-analysis/fsnlp/tagger.en.html
[23] Helmut Schmid and Sabine Schulte im Walde, Institut für Machinelle Sprachverarbeitung (IMS), University of Stuttgart.

terms of what phrases are subparts of other phrases. This information is often depicted in a tree form (figure 6.1).

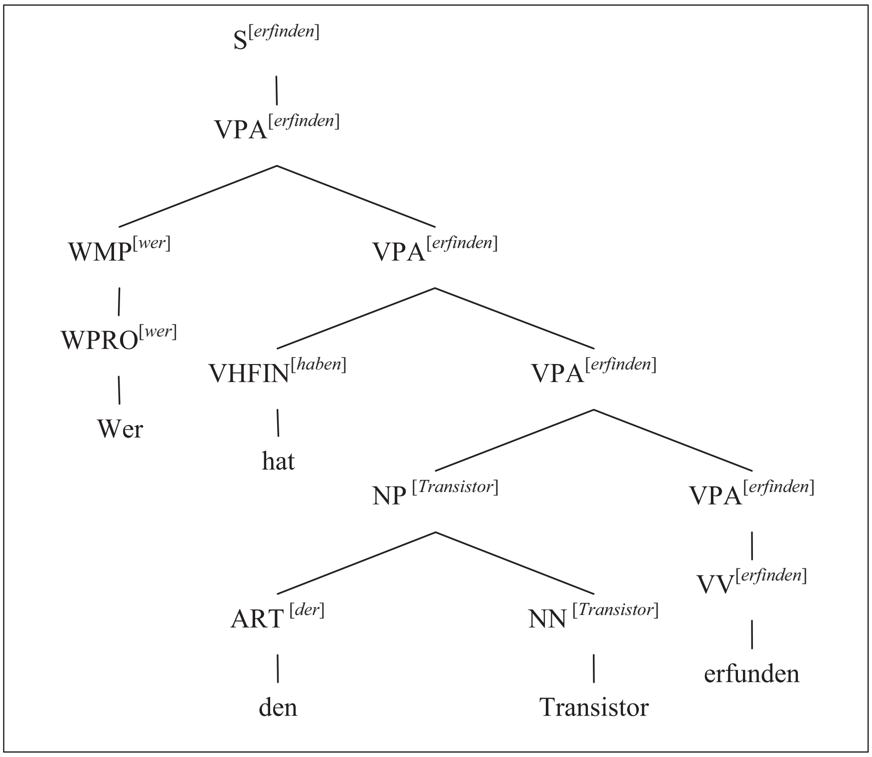


Figure 6.1: Example of a parse tree for the sentence "Wer hat den Transistor erfunden" (Who has invented the transistor).

The linguistic pre-processing is performed with a *parser*. There are only few German parsers[24] that could be used in our project; we retained *LoPar*[25]. LoPar was created by the same authors who developed TreeTagger (see strategy 2), and we received the same support to modify the lexicographical sources. LoPar comprises parsing with a *head-lexicalized probabilistic context-free grammar* (HL-PCFG).

Strategy 3 was implemented prototypically in CHESt v4 (see chapter 9), and published in [LM07], but it was never evaluated in a benchmark test or an assessment; the reason is explained below.

On the one hand, this NLP strategy considerably improved the translation of the user query into a logical form. On the other hand, it turned out that it will fail in a real-life scenario. First, a strict syntax analysis of the user question is not possible when the users are free to enter questions as they want. This is especially true if the users are students and do not master the language. User experiments (see chapter 11) confirmed other authors, e.g., [Blo01, Mor05] that students express rarely in a very precise way, and make spelling and grammatical errors. Therefore, a syntax parser will have difficulties to build meaningful syntax trees out of erroneously user questions. Similar statements were made in [Fra96, DBB+02, RRFD04]. Secondly, the quality of the retrieval relies not only on the preciseness of the NLP, but also on the quality of the semantic annotation of the documents in the knowledge base (see section 5.3). This means that the more precisely the user

[24]At the time we were working on the prototype CHESt that had a German knowledge base (see chapter 9).
[25]`http://www.ims.uni-stuttgart.de/projekte/gramotron/SOFTWARE/LoPar-en.html`

question is translated into a logical form, the more ambiguities are created, and the more documents will be yielded. To overcome this problem, the users must be forced to enter very detailed and well formulated questions, which is rather unrealistic.

6.3.4 Strategy 4

Based on the experiences with the above NLP strategies, we were searching for the best possible compromise. The solution is *partial parsing*, also called *shallow parsing* or *light parsing* [All94, Mit04, MS99]. Shallow Parsing is a NLP technique that attempts to provide machine understanding of the structure of a sentence, but without parsing it fully into a parsed tree form. Shallow parsing most often refers to the task of *chunking*. The output of a chunker is a series of words that together represent a grammatical unit (mostly either noun, verb, or preposition phrase, with less-frequent occurrences of adverb, adjective, clause, and other phrases). The output is different from that of a fully parsed tree, because it consists of series of words that do not overlap and that do not contain each other. This makes chunking an easier NLP task than full parsing.

Two reasons led us to built our own chunker. First, we did not find a partial parser for German that we could include in our system as easily as TreeTager or LoPar; existing tools often lack of specific domain words. Secondly, external programs like TreeTagger or LoPar are executed in a shell. Starting an external tool like TreeTagger takes on average 5 seconds, which is way too long for an ergonomic, interactive system.

The NLP module of our most advanced E-Librarian Service is based on a chunker, the computation of word equivalences, and the semantic interpretation of the identified relevant words in the user question. This strategy will be depicted in detail in the next section.

6.4 Natural Language Processing in our E-Librarian Service

In this section, we will describe how the NLP strategy 4 (see section 6.3.4) was implemented. We will start with a description of the domain language and dictionary, then we will explain the solutions that we explored to compute word equivalences. The major contribution in this section will be our technique for the semantic interpretation of a user question. Finally, we will present the multi-language feature of our NLP module.

6.4.1 Domain Language and Dictionary

NLP systems use dictionaries, thesauri, or other structured or unstructured repositories of NL words. In many related projects (see section 5.1) an existing knowledge source was used, usually *WordNet*[26] [Ma98].

WordNet is a lexical database for English based on psycholinguistic principles. Its information is organized in units called "synsets", which are sets of synonyms that are interchangeable in a particular context, and are used to represent different meanings. WordNet contains a set of pairs (w, s), where w is a string of ASCII characters, and the meaning s is an element of a set of meanings or synset. Most of the synsets are accompanied by explanatory glossaries, and they are all organized in a network by means of semantic relationships of the type:

- *hyperonym*: a word with a more general meaning (e.g., animal is a hyperonym of cat),
- *hyponym*: a word with a more specific meaning (e.g., cat is a hyponym of animal),

[26] `http://wordnet.princeton.edu/`

- *synonym*: a word with identical meaning (e.g., car and automobile are synonyms),
- *homonyms*: words with identical spelling but different meaning (e.g., Ada is a programming language but also a person).

We investigated how useful *GermaNet* (the German version of WordNet) could be for our project. First, GermaNet is not dedicated to a domain. Like other large scale dictionaries, GermaNet lacks specific domain expressions on the one hand, but on the other hand contains too much knowledge about other domains. This increases the problem of ambiguous interpretations for a given word. For instance, WordNet returns three interpretations for the noun "Pascal": a unit of pressure, the name of a French mathematician, and the name of a programming language. Only the latter would be interesting in the context of computer science. Secondly, a GermaNet license requires major financial investments.

We decided to create our own dictionary — based on an appropriate domain ontology — that uses all relevant words of the domain, and not too many of other words. The creation of the dictionary is described in section 5.3. The result is a rich and structured domain dictionary for our E-Librarian Service, which covers a domain language that may or may not contain all the possible words used by the user. We now give a more formal description for the domain language and dictionary.

Definition 10 (domain language) *Let L be the set of all existing words over a certain alphabet that might be used to formulate a query so that $L \subseteq \Sigma^*$. A domain language L_H is the set of all words that are known in a given domain — we will call them the well-known-words (wkw) in the remaining part of the document — so that $L_H \subseteq L \subseteq \Sigma^*$.*

A domain language can contain verbs, nouns, articles as well as names, numbers, etc. Furthermore, the same domain language can contain words in different languages. The domain language gives no semantics or other description of its elements; it is just a set of stored words. The semantics are attached to each word by classification in a domain dictionary, which is structured in a WordNet-like hierarchical way as described above.

Definition 11 (domain dictionary) *A domain dictionary $H = (V, E, v_0)$ is a rooted and oriented tree, where each node except the root-node (v_0), has one or more parents. E is the set of all edges and V is the set of all nodes (vertices) with $V = \{(s,T)|s \in S\}$, where s is a unique label (see definition 12) and T is a set of wkw (see definition 10) associated to a node; $T = \{t|t \in L_H\}$.*

Concept: TCPIP
$s =$ `chest:TCPIP` $T = \{$TCP/IP, TCPIP, TCP-IP$\}$

Figure 6.2: Example of a node in the domain dictionary.

A node represents a concept in an ontology. The words that refer to this concept are regrouped in T. We assume that each set of words T_i is semantically related to the concept that the node v_i represents. Figure 6.2 illustrates this idea with the concept TCPIP according to the taxonomy shown in figure 5.1. Here, the words "TCP/IP", "TCPIP", and "TCP-IP" refer to the same concept TCPIP.

Of course, a certain term, e.g., "Ada" could refer to different concepts; "Ada" is the name of a programming language but also the name of a person (Augusta Ada Lovelace). We will detail this problematic in section 6.4.3.2.

Not all words in L_H must be associated with a concept. Only words that are semantically relevant are classified. In general, nouns and verbs are best indicators of the sense of a question [Kup93]. The difference between words that are semantically irrelevant and words that are not contained in L_H is that, for the second ones, the system has absolutely no idea if they are relevant or not.

Definition 12 (label) *A label is a unique identifier for a concept in a domain dictionary so that for a given label s one can find the corresponding concept and vice versa. S is the set of all existing labels in a domain dictionary.*

Technically, the kind of labels used depends on the encoding framework for annotating the documents in the knowledge base (see section 5.3). In our case, a label is a namespace prefix (e.g., `chest`) and a local name (e.g., `EComponent`). Together they form the label of a node (e.g., `chest:TCPIP`) like illustrated in figure 6.2.

Definition 13 (classification of documents) *Let D be the set of all documents in the knowledge base, then a document $d \in D$ is classified under a concept k if d is about k, and there is not a more specific concept k' under which d could be classified.*

In certain cases, a document can be classified in more than one concept. For instance, the document introducing the protocol TCP/IP is classified in a concept named TCPIP but also in a concept named Protocol-Suite.

6.4.2 Word equivalence

Common in all four NLP-strategies described in section 6.3 is the task of computing the similarity of words. In fact, the user can make spelling errors, e.g., "Who *invXented* the transistor?". Here, the NLP module of our E-Librarian Service must compute the best equivalent word from the domain dictionary for the chunk "*invXented*".

Definition 14 (word equivalence) *The function $\pi(a, b)$ quantifies the similarity of two given words ($a, b \in L$) using a logic W, so that a and b are said to be equivalent w.r.t. to a given tolerance ε, written $a \equiv b$, iff $\pi(a, b) \leq \varepsilon$.*

The choice of W depends on how expressive one wants to be in the approximation of the meaning of the concepts, and on the complexity of the NLP techniques used to process words. We explored two solutions.

6.4.2.1 Solution 1

In a first and now abandoned attempt described in [LM04b], we represented the words as a tree, where each node represents a character. This approach based on graph theory allowed to compare two words node by node, and to compute the length of their equal *trunk*, as well as their remaining unequal *tail*.

However, this solution fails when the difference starts very early, e.g., "iXvented" and "invented" will only have 1 (the first) character in common, and a completely different tail. Thus, their computed difference is very high, and they will not be considered as being equivalent.

6.4.2.2 Solution 2

The above solution was replaced by a more efficient approach: the *Levenshtein function*, also called *edit distance* [CEE+01]. For two given string arguments a and b, the function computes the number of deletions, insertions, or substitutions required to transform a into b. The greater the Levenshtein distance, the more different the strings are. For example, the Levenshtein distance for "iXvented" and "invented" is 1, because one modification must be done to turn "iXvented" into "invented". Thus both words are very related.

The main disadvantage of this solution is that each word must be stored in nearly all of its morphological forms in the dictionary. But it turned out that the vocabulary employed by the users is only a very limited set of words. The domain dictionary for the different prototypes contains not more than 3000 words each.

6.4.3 Semantic interpretation

The representation of context-independent meaning is called the *logical form*. The process of mapping a sentence to its logical form is called *semantic interpretation* [All94]. The semantic interpretation in our E-Librarian Service is the translation of a NL question into a logical form using a given strategy (see section 6.3). We decided to use Description Logics (DLs) as knowledge representation language (see chapter 3). First, DLs have the advantage that they come with well defined semantics and correct algorithms [BCM+03]. Secondly, the link between DLs and NL has already been established [Sch93, Fra03]. Third, as our E-Librarian Service is based on Semantic Web technologies like OWL (see chapter 5), it seems evident that the same formalism, i.e., OWL DL, should be used throughout the complete project. Indeed, DLs are the common knowledge representation language for the knowledge base encoded in OWL DL, and the semantically interpreted user question.

For example, the semantic interpretation of the sentence "What are the tasks of TCP/IP?" would result in the logical form, i.e., DL-concept description: $\mathsf{TCPIP} \sqcap \exists \mathsf{hasTask}$.

A core part of the semantic interpretation in our E-Librarian Service is a mapping algorithm. This step maps each word from the user question to one or more ontology concepts and roles, and resolves the arguments of each role. The second step is to resolve possible ambiguities, i.e., the semantic interpretation of multiple-sense words. The final step is the generation of a DL-concept description that represents the meaning of the complete user question.

6.4.3.1 Step 1: Mapping of Words

Definition 15 (user question) *A user question q is a sentence that the user formulates in a language L, and which is composed of words so that $q = \{w_1, ..., w_n\}$ with $n \geq 1, w_k \in L$, and $k \in [1..n]$.*

Definition 16 (mapping) *The meaning of each word $w_k \in L$ is made explicit with the mapping function $\varphi : L \rightarrow V$ over a domain language L_H w.r.t. a domain dictionary $H = (V, E, v_0)$, so that $\varphi(w_k)$ returns a set of interpretations Φ defined as follows,*

$$\Phi = \varphi(w_k) = \{v_i | \exists x \in ft(v_i) : w_k \equiv x\}.$$

The function $ft(v_i)$ returns the set of wkw T_i associated to the node v_i (see definition 11), and $w_k \equiv x$ are two equivalent words (see definition 14).

Definition 17 (semantic relevance) *A word w_k is semantically relevant if there exists at least one concept in the domain dictionary H to which w_k can be mapped, so that $\varphi(w_k) \neq \emptyset$.*

The mapping function φ is used for the semantic interpretation of a L-word w, so that $\varphi(w)$ returns a set of valid interpretations, e.g., φ("TCP-IP") $=\{\mathsf{TCPIP}\}$. In this way, semantically irrelevant words are filtered out, e.g., $\varphi(\text{“the”}) = \{\}$.

The system allows a certain tolerance regarding spelling errors, e.g., the word "comXmon" will be considered as "common", and not as "uncommon". Both words, "common" and "uncommon", will be considered for the mapping of "comXXmon". In that case, the mapping function will return two possible interpretations. We will see below how such an ambiguity is resolved.

6.4.3.2 Step 2: Resolving Ambiguities

It is possible that a word can be mapped to different concepts at once, so that $|\Phi| > 1$. We introduced in [LM06c] the notion of *focus* to solve this problem. The focus is the function f which returns the best interpretation for a given word in the context of the complete user question.

Definition 18 (focus) *The interpretation of a mapping $\varphi(w_k)$ in the context of a given question q is made explicit by the function f. The focus, written $f_q(\varphi(w_k)) = v'$, guarantees the following:*

1. *$v' \in \varphi(w_k)$; the focused word is a valid interpretation,*
2. *$|f_q(\varphi(w_k))| = [0, 1]$; the focus function returns 0 or 1 result,*
3. *$\top \leq v' \leq \bot$, iff $f_q(\varphi(w_k)) \neq \emptyset$; if the focusing is successful, then the word is semantically related to the ontology's domain,*
4. *$(\exists x \in ft(v'), \forall y \in ft(v_i \in \varphi(w_k)))\ \pi(w_k, x) \geq \pi(w_k, y)$, iff $f_q(\varphi(w_k)) \neq \emptyset$; if the focussing is successful, then the returned interpretation contains the best matching word of all possible interpretations.*

Let us consider as illustration the word "Ada" which is called a multiple-sense word. In fact, in the context of computer history, "Ada" can refer to the programming language named "Ada", but it can also be the name of the person "Augusta Ada Lovelace". The correct interpretation can only be retrieved accurately by putting the ambiguous word in the context of a complete question. For instance, the context of the sentences "Who invented Ada?" and "Did the firms Bull and Honeywell create Ada?" reveals that here, "Ada" is the programming language and not the person "Ada".

The focus function relies on the role's signature. A role r has the signature $r(s_1, s_2)$, where s_1 and s_2 are labels (see definition 12). The signature of each role defines the kind of arguments that are possible. Hence, wasInventedBy(Thing,Creator) is the role $r =$wasInventedBy that has the arguments $s_1 = \mathsf{Thing}$ and $s_2 = \mathsf{Creator}$. Technically, the signature of each role is defined in the ontology using the RDFS-elements `range` and `domain`. The following mappings are computed for the question $q =$ "Who invented Ada?":

$$\begin{aligned} \varphi(\text{“Who”}) &\rightarrow \{\mathsf{Creator}\} \\ \varphi(\text{“invented”}) &\rightarrow \{\mathsf{wasInventedBy(Thing,Creator)}\} \\ \varphi(\text{“Ada”}) &\rightarrow \{\mathsf{Person,\ Language}\} \end{aligned}$$

The system detects an ambiguity for the word "Ada", which is mapped to an instance of the concept Person, but also to an instance of the concept Language. The focus function computes the following combinations to resolve the ambiguity[27]:

[27] As usual in computational linguistics, we mark with an * the sentences that are not grammatically correct.

1. Was Ada invented by who?* $\rightarrow$ $\mathsf{wasInventedBy}$(“Ada”,“Who”)
2. Was Ada invented by Ada? $\rightarrow$ $\mathsf{wasInventedBy}$(“Ada”,“Ada”)
3. Was who invented by Ada?* $\rightarrow$ $\mathsf{wasInventedBy}$(“Who”,“Ada”)
4. Was who invented by who?* $\rightarrow$ $\mathsf{wasInventedBy}$(“Who”,“Who”)

Cyclic combinations like (2) and (4) are not allowed. As for (3), it does not match the role's signature because $s_1 = \mathsf{Creator}$ (“Who”), but Thing is required. As for (1), s_1 can be Person or $\mathsf{Language}$ (“Ada”). The role's signature requires Thing, therefore Person is excluded as a valid interpretation because $\mathsf{Person} \not\sqsubseteq \mathsf{Thing}$. As $\mathsf{Language} \sqsubseteq \mathsf{Thing}$, a valid interpretation is found, and in the context of this question the word “Ada” refers to the programming language “Ada”. Finally, the result of the focus function is:

$$f_q(\varphi(\text{“Ada”})) = \mathsf{Language}.$$

Indeed, (1) represents the question “Who invented Ada?”. It is still possible that the focus function cannot resolve an ambiguity, e.g., a given word has more interpretations but the focus function returns no result. In such a case, the system will generate a DL-concept description (see section 6.4.3.3) for each possible interpretation in Φ. Based on our practical experience we know that users generally enter simple questions, where the disambiguation is successful.

6.4.3.3 Step 3: Generation of a DL-Concept Description

Definition 19 (semantic interpretation) *The semantic interpretation of a user question q is the translation of each linguistic clause into a DL-concept description w.r.t. a given ontology, written:*

$$Q = \prod_{k=1}^{n} f_q\left(\varphi(w_k \in q)\right)$$

where n is the number of words in the sentence (see definition 15).

For instance, the question “Who invented Ada” will be translated into the DL-concept description:

$$Q \equiv \mathsf{Ada} \sqcap \; \exists \mathsf{wasInventedBy},$$

and not in $Q \equiv \exists \mathsf{wasInventedBy.Ada}$ — representing, e.g., the question “What was invented by Ada?” — because the ambiguity initiated by the word “Ada” was correctly resolved.

Our NLP module guarantees that the output of the semantic interpretation is “optimized”, i.e., the generated DL-concept description is in a normal form. There are different normal forms like $\perp$-*normal form* or the *Reduced Concept Description* (RCD). For the rest of this document, we will always refer to the normal form by its RCD. The RCD says that a concept description does not contain any redundant information.

Definition 20 (reduced concept description) *Let $C \equiv A_1 \sqcap ... \sqcap A_n$ be a DL-concept description and $A_i, i \in [1..n]$ are clauses in C. C is reduced if either $n = 1$ or no clause in C subsumes the conjunction of the other clauses:*

$$\forall 1 \leq i \leq n : A - A_i \not\sqsubseteq A_i.$$

6.4.3.4 Limitations and Constraints

The complexity of the semantic interpretation depends directly on the DL-language used (see figure 3.2). In our implementation of the semantic interpretation we rely on the DL-sublangauge $\mathcal{EL}$, thus we do not consider negations, e.g., "Who did *not* invent Ada?". Furthermore, our NLP module is not able to correctly process questions that are composed of more sentences, e.g., "What are the tasks of TCP/IP and who invented Ada?". This is mainly due to the fact that our employed NLP strategy (see section 6.3.4) does not perform a full syntactic analysis of the user question. Finally, a problem that is not yet solved are *inter-clausal dependencies*. For instance, "Is the person who invented TCP/IP also the person who invented Ada?". This kind of complex sentence — even if it is very improbable that a student would formulate such a question — cannot be processed correctly in the current state of our E-Librarian Service.

6.4.4 Multiple-Language Feature

A very useful feature of our NLP module is its independence with regard to language. By simply changing the domain dictionary, e.g., a German dictionary instead of a French one, the complete E-Librarian Service can be used — even without any recompilation — for another language. This was very useful in our experiments (see chapters 11 and 12), where different knowledge bases in different languages were used.

Also, the domain dictionary can be mixed. In such a case, the E-Librarian Service understands different languages simultaneously. For instance, the domain dictionary knows that the words "erfinden", "inventer" and "to invent" all refer to the same role: wasInventedBy.

Chapter 7

Semantic Information Retrieval

Contents

The core part of our E-Librarian Service is a multimedia information retrieval module that performs a semantic search over the knowledge base. It retrieves only few but semantically pertinent documents as an answer to the users' questions.

In this chapter, we will describe how our multimedia information retrieval module works. We will start in section 7.1 with a short overview of multimedia information retrieval in computer science. Then, we will summarize our contributions in section 7.2. Three different retrieval strategies that were explored will be presented in section 7.3. Section 7.4 will describe how the best answer according to a query is computed, i.e., how the semantic difference between "candidate documents" and the query is derived. We will conclude the chapter with an illustrating example in section 7.5.

7.1 Multimedia Information Retrieval in Computer-Science

Multimedia information retrieval (MIR) constitutes a very active multidisciplinary research area. It is being transformed into a cross-cutting field. Extending beyond the borders of culture, art, and science, the search for digital information is one of the major challenges of our time. Digital libraries, bio-computing & medical science, the Internet, streaming video, databases, cultural heritage collections, and peer-2-peer networks have created a worldwide need for new paradigms and techniques on how to browse, search, and summarize multimedia collections. For this reason, multimedia information systems are widely recognized to be one of the most promising fields in the area of information management.

The most important characteristic of a MIR system is the variety of data it must be able to support. Multimedia systems must have the capability to store, retrieve, transport, and present data with very heterogeneous characteristics such as text, images (both still and moving), graphs, and sound. For this reason, the development of a multimedia system is considerably more complex than a traditional information system. Conventional systems only deal with simple data types, such as strings or integers. On the contrary, the underlying data model, the query language, and the access and storage mechanisms of a multimedia system must be able to support objects with a very complex structure. The need then arises for developing MIR systems specifically for handling multimedia data.

Traditional information retrieval systems only deal with textual, unstructured data; therefore, they are unable to support the mix of structured and unstructured data, and different kinds of media, typical of a MIR system [BYRN99]. Hence, a traditional information retrieval (IR) system does not support metadata such as that provided by database schema, which is a fundamental component in a database management system (DBMS). On the other hand, multimedia applications need to structure their data at least partially. However, the notion of schema may need to be weakened w.r.t. the traditional notion to ensure a higher degree of flexibility in structuring data. Moreover, a MIR system requires handling metadata, which is crucial for data retrieval, whereas traditional IR systems do not have such requirement.

7.2 Objective and Contributions

Our E-Librarian Service is an ontology driven semantic search engine over a multimedia knowledge base that covers a given domain (e.g., computer history, or fractions in mathematics). Its task is to retrieve only semantically pertinent documents from the knowledge base w.r.t. a given user query. Among all the documents in the knowledge base that have some common information with the user query, our algorithm is able to identify the most pertinent match(es), keeping in mind that the user in general expects an exhaustive answer while preferring a concise answer with only little or no information overhead.

Our E-Librarian Service can be perceived as a specialization of *passage retrieval* techniques. Passage retrieval techniques have been extensively used in information retrieval settings, and have proven effective for document retrieval when documents are long, or when there are topic changes within a document [LC02, RCPB04].

Our MIR algorithm is based on the *concept covering problem* in Description Logics (DLs); it computes the semantic distance between the query and the documents in the knowledge base, and yields only those most related semantically. The ranking of the documents is computed according to the semantic distance between the query and the matching documents. Benchmark tests (see chapter 8) will prove the quality and reliability of our solution.

7.3 Explored Strategies

In this section, we will present the three MIR strategies that we explored. Their common specifications are the following: the input is a query and an ontology, and the output is a set of references (URIs) to the resulting documents.

7.3.1 Strategy 1

The first retrieval strategy that we explored was based on two principles; first, the mapping of a user question to a *general assertion*, and second, the generation of a *semantic query* in a certain query language.

As for the first issue, it was based on the fact that all systems that interact with humans have some concepts about their users [CEE$^+$01]. If the system is dedicated to a certain group of users, e.g., students in a mathematics lesson, then predications about the nature of their questions can be made. In this way, a set of possible assertions can be created to generalize and to regroup all imaginable user questions. The strategy consists in abstracting the user question until it can be mapped to the best matching general assertion known by the system. For example, the question "Who invented TCP/IP?" would be mapped to the general assertion: "Something was invented by someone". Then, the general assertion would be transformed into the form: "TCP/IP was invented by X", with the aim to retrieve a matching document for X. It turned out that we needed only few general assertions for each domain.

As for the second issue, a semantic query is generated based on the identified general assertion and the values from the original user query. The latter reveal the nature of the sentence, e.g., number of verbs and objects in the sentence. A short set of rules allowed to create a semantic query, e.g., rule 1: no verb in the sentence, rule 2: one object in the sentence, and rule 2: two objects in the sentence. As for the above example, one verb (to invent) and two objects (TCP/IP and X) were identified, where X is the missing value, and the according rule is fired. We used the RDF query language RDQL[1] [MSR02] to generate a semantic query, also called ABox-query.

Depending on the complexity of the user question, a semantic query can be split into several sub-queries. In that case, the yielded results are optimized in the way that preference is given to documents which are found in all sub-queries.

Most of this solution was published in [LM04b, LM05a]. It seems evident that the major weakness of this MIR strategy is that the quality of the retrieval is directly related to the available general assertions.

7.3.2 Strategy 2

We decided to translate the user question into a logical form to infer over the knowledge base, and to overcome in this way the limitations of the general assertions of strategy 1. Furthermore, the nature of the question (*open* or *close*) reveals the missing part. An *open question* contains a question word, e.g., "*Who* invented TCP/IP?", whereas a *close question* (logical- or yes/no question) does not have a question word, e.g., "Did Vinton Cerf contribute to the invention of TCP/IP?". As for the first kind of questions, the missing part — normally not an individual but a concept — is the subject of the question, and therefore the requested result. The result of the query is the set of all models $\mathcal{I}$ in the knowledge base $\mathcal{K}$. As for the second kind of questions, there is no missing part. Therefore, the answer will be "yes" if $\mathcal{K} \models Q$, otherwise it is "no".

[1] Recently being substituted by SPARQL (see section 2.2).

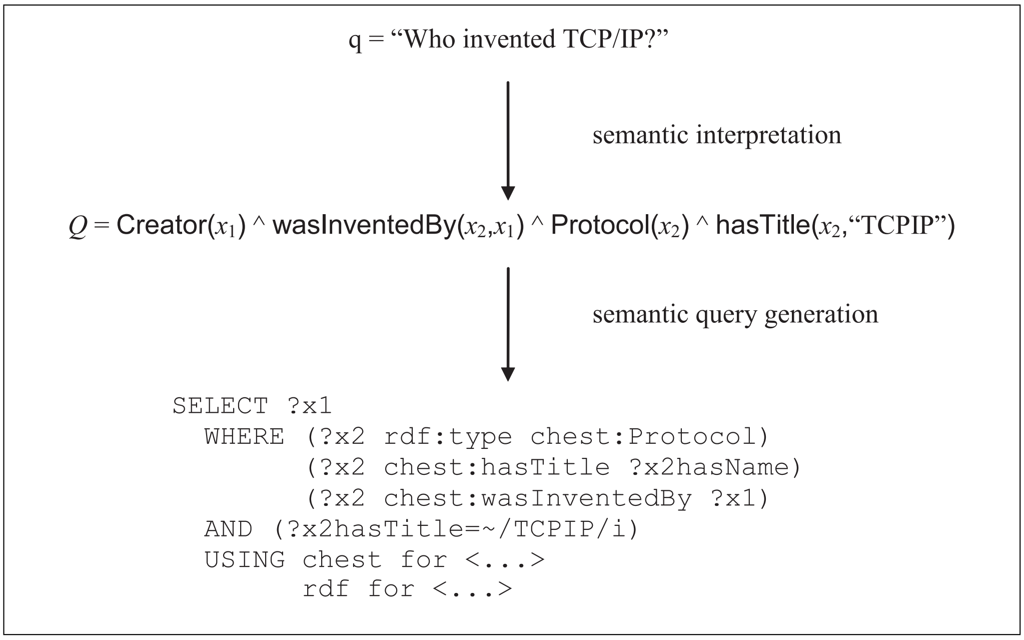


Figure 7.1: Example of the translation of a user question into a RDQL query.

An illustration of this strategy is shown in figure 7.1. Most of this solution was published in [LM05b, LM06c, LM06b, LME06, LM07].

We tested different external reasoners like Fact[2] [Hor98], Fact++[3] [BVL03], Racer [HM01], and Pellet[4] [SP04]. However, they support only standard interference services like subsumption and satisfiability. Indeed, computing the best matching documents w.r.t. to a query requires more advanced reasoning capabilities.

7.3.3 Strategy 3

We studied three different approaches related to document matching and retrieval based on non-standard inferences in DLs. First, an approach for matching documents [Küs01] (see section 3.2.2.5). It introduces a matching problem modulo equivalence and modulo subsumption as being of the form $C \equiv^? D$ and $C \sqsubseteq^? D$ respectively, where C is a description and D a pattern. A *matcher* of these problems is a substitution σ such that $C \equiv \sigma(D)$ and $C \sqsubseteq \sigma(D)$, respectively. The solution is based on computing homomorphisms between description trees. Although this is an excellent solution for dealing with complex concept descriptions such as for comparing complete documents, it is less appropriate for our purpose. In our case, documents in the knowledge base are described by simple semantic annotations with few role-imbrications. The resulting description trees are rather flat and comprise rarely more than two levels.

Second comes the concept covering problem [HLRT02, BHyP$^+$06] (see section 3.2.2.7). It is based on DLs with structural subsumption. The proposed algorithm for identifying the best covers relies on the computation of minimal transversals in a hypergraph [KS05]. Similar approaches have already been explored in the field of artificial intelligence in the 1980s as so called "Truth

[2] `http://www.cs.man.ac.uk/~horrocks/FaCT/`
[3] `http://owl.man.ac.uk/factplusplus/`
[4] `http://www.mindswap.org/2003/pellet/`

Maintenance Systems" (TMS), see e.g., [dK86, MJ89, BH90]. The concept covering problem is very pertinent for our E-Librarian Service because it always finds the best cover, i.e., the best matching documents. It will be discussed in greater detail in section 7.4.

Another definition of the concept covering problem that eliminates the limitation of DLs to provide structural subsumption has been presented in [CNS+05a] (see section 3.2.2.6). There, the concept covering problem is based on the concept abduction problem (CAP), which is able to provide an explanation if subsumption does not hold. It is stated as follows: S (supply) and D (demand) are two descriptions in a DL $\mathcal{L}$, and satisfiable in a terminology $\mathcal{T}$. A CAP, identified by $< \mathcal{L}, S, D, \mathcal{T} >$, is finding a concept $H \in \mathcal{L}$ (hypotheses) such that $S \sqcap H \sqsubseteq_{\mathcal{T}} D$, and moreover $S \sqcap H \not\equiv \bot$. One of the weaknesses of this solution is that it does not always return an optimal cover.

We decided to base our MIR strategy on the concept covering problem as presented in [HLRT02]. First, it is an efficient solution for finding semantically pertinent documents by inferring over the knowledge base. Secondly, DLs with structural subsumption provide sufficient expressiveness for our E-Librarian Service. Thirdly, it always returns a correct and optimal answer. Finally, the solution is simple and adapted to the rather simple semantic annotation of our documents. Our modified solution of the concept covering problem is described in detail in section 7.4.

7.4 Semantic Information Retrieval in CHESt

By *retrieval* we refer to the idea of answering a user's question by identifying only the semantically most pertinent documents according to a given question. In addition, the MIR module must quantify the quality of the yielded result(s), i.e., to measure the semantic distance between the user's query and the identified documents. This measure is also used to rank similar results.

Our solution is based on the *concept covering problem* and on the quantification of the *semantic distance*. It always proposes a solution, even if the system concludes that there is no exhaustive answer. By quantifying the missing and supplementary information, the system is able to compute and to visualize the quality and pertinence of the yielded document(s). Our solution was published in [LSM07a, LSM07b].

In this section, we will first describe our solution for finding pertinent documents. Secondly, we will explain how the semantic distance between a query and a set of "candidate documents" is computed. Then we will describe how the best matching document(s) are identified. Finally, we will present the algorithm LOFind as implementation of our solution.

7.4.1 Finding Pertinent Documents

The original concept covering problem defines a *cover* of a concept C w.r.t. a terminology $\mathcal{T}$ as being the conjunction of some defined concepts in $\mathcal{T}$ that share some information with Q (see section 3.2.2.7).

Although the principle of this solution is pertinent for our E-Librarian Service, we think that a user might not be satisfied if the delivered answer to his/her precise question is a concatenation of different — normally not related — documents from the knowledge base. First, there is no transition between the different documents. Secondly, we risk that there is way too much information in the delivered answer, because the original algorithm adds all necessary and (partially) matching documents to the answer until the query is covered completely.

We learned from experiments (see chapter 11) that users prefer few but precise answers — even if these answers are not complete — rather than a set of different concatenated documents. This assertion is confirmed by pedagogical analysis, e.g., [LH99, FDD+99, HS00, Blo01], which conclude that students are searching for one — the best — answer, and prefer to reformulate their query to reduce the number of results.

Our modified concept covering problem defines a cover (candidate document) as being a concept description C w.r.t. a terminology $\mathcal{T}$ that shares some information with another concept description Q (query) w.r.t. $\mathcal{T}$.

Definition 21 (cover) *Let $\mathcal{L}$ be a DL with structural subsumption, $\mathcal{T}$ be an $\mathcal{L}$-terminology and $C_{\mathcal{T}} = \{C_i \not\equiv \perp, i \in [1,n]\}$ the set of concept descriptions occurring in $\mathcal{T}$. Then $C_j \in C_{\mathcal{T}}$ is a cover of a $\mathcal{L}$-concept description $Q \not\equiv \perp$ if $Q - lcs_{\mathcal{T}}(Q, C_j) \not\equiv Q$, where the operator "$-$" represents the semantic difference.*

7.4.2 Computing the Semantic Distance

To find the best matching document among all candidates, we refer to the notion of semantic distance (or semantic relatedness); the smaller the semantic distance between the query and the candidate document, the more pertinent the document is for the user. Different alternative approaches exist, e.g., [MBR01, dFE06, BWH05, BH06, KEW01].

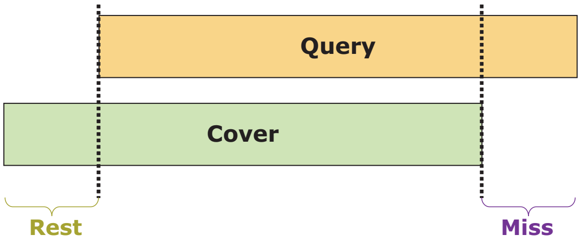


Figure 7.2: Graphical illustration of the Miss and Rest.

The best cover can be defined based on the remaining information in the query (denoted as $Miss$) and in the cover (denoted as $Rest$). The Miss is the part of the query that is not part of the cover, and the Rest is the information that is part of the cover but not required by the query. An illustration is given in figure 7.2.

Definition 22 (miss and rest) *Let Q, C be be two $\mathcal{L}$-concept descriptions, then:*

- *the Miss of Q w.r.t. C, denoted as $Miss(Q,C)$ is defined as follows:*
  $Miss(Q,C) = Q - lcs_{\mathcal{T}}(Q,C)$,
- *the Rest of Q w.r.t. C denoted as $Rest(Q,C)$ is defined as follows:*
  $Rest(Q,C) = C - lcs_{\mathcal{T}}(Q,C)$.

The Miss and Rest are $\mathcal{L}$-concept descriptions. To quantify them in terms of integers, we need to measure the size of a DL-concept description.

Definition 23 (size of a concept description) *The size of a DL-concept description, denoted as $|\cdot|$ is inductively defined by:*

- $|\perp| = |\top| = 0$,
- $|A| = |\neg A| = 1$,

- $|\exists r.C| = |\forall r.C| = 2 + |C|$,
- $|C \sqcap D| = |C \sqcup D| = |C| + |D|$,
- $|\neg C| = |C|$.

Hence, the size of the concept description $Q \equiv \mathsf{TCPIP} \sqcap \mathsf{Protocol} \sqcap \exists \mathsf{hasTask}$ — representing the question: "What are the tasks of the protocol TCP/IP?" — is computed as follows:

$$\begin{array}{rcl} |\mathsf{TCPIP}| & = & 1 \\ |\mathsf{Protocol}| & = & 1 \\ |\exists \mathsf{hasTask}| & = & 2 \\ \hline |Q| & = & 4 \end{array}$$

7.4.3 Identifying the Best Matching Document(s)

The best cover can be assumed as being the cover with the smallest Miss and Rest.

Definition 24 (best cover) *Let C, D be two $\mathcal{L}$-concept descriptions. A cover C is called a best cover w.r.t. Q using a terminology $\mathcal{T}$ iff:*

- *C is a cover w.r.t. Q using $\mathcal{T}$, and*
- *there does not exist any cover C' of Q using $\mathcal{T}$ such that*

$$(|Miss(Q, C')|, |Rest(Q, C')|) < (|Miss(Q, C)|, |Rest(Q, C)|)$$

where $<$ stands for the lexicographic order.

By choosing a lexicographical order we give preference to a minimized Miss. For example, for (Miss,Rest), the couple (1,2) < (2,1) because the first couple has a smaller Miss than the second one. In fact, the E-Librarian Service aims to give an exhaustive answer in the first place, i.e., to yield an answer that covers the user's query as much as possible, even if there is more information in the answer than required. Only in the second place, the Rest is considered in order to rank the results that have the same Miss.

7.4.4 Algorithm for the Retrieval Problem

Our algorithm to compute the best cover is called LOFind (see figure 7.3). As input, a query Q is expected that was translated into a $\mathcal{L}$-concept description (see chapter 6), and a $\mathcal{L}$-terminology $\mathcal{T}$, i.e., a set of semantic descriptions of documents (see section 5.3). The output of LOFind is the set E of best covers w.r.t. Q using $\mathcal{T}$.

The algorithm works as follows. Let us suppose that $C_{\mathcal{T}}$ is the set of semantic descriptions of the documents in the knowledge base. Then, each document is tested if it is a cover (line 4). If so, then it will only be maintained, if either the size of its Miss is smaller than (line 5), or equal to (line 8) the smallest Miss found up to now. In the first case, the current document replaces all the former best cover-candidates (lines 6 + 7). In the second case, the current document is added to the best cover-candidates found up to now (line 9).

Require: a query $Q \not\equiv \perp$, a set of concept descriptions $C_\mathcal{T} = \{C_i \not\equiv \perp, i \in [1, n]\}$
Ensure: a set of best covers $E = \{C_j \in C_\mathcal{T}, j \in [0..n]\}$
$E \leftarrow \emptyset$
$MinMiss \leftarrow +\infty$
for each $C_i \in C_\mathcal{T}$ **do**
  if $Q - lcs(Q, C_i) \not\equiv Q$ **then**
    if $|Miss(Q, C_i)| < MinMiss$ **then**
      $E \leftarrow C_i$
      $MinMiss \leftarrow |Miss(Q, C_i)|$
    else if $|Miss(Q, C_i)| = MinMiss$ **then**
      $E \leftarrow E \cup C_i$
    end if
  end if
end for

Figure 7.3: The algorithm LOFind.

$\mathsf{LO}_1 \equiv \mathsf{Protocol}$
$\mathsf{LO}_2 \equiv \exists\mathsf{howWorks} \sqcap \mathsf{TCPIP}$
$\mathsf{LO}_3 \equiv \mathsf{Protocol} \sqcap \exists\mathsf{hasTask.ErrorHandling}$
$\mathsf{LO}_4 \equiv \mathsf{Protocol} \sqcap \exists\mathsf{hasTask.FlowControl}$
$\mathsf{LO}_5 \equiv \mathsf{FlowControl}$

Figure 7.4: Example of a terminology of LO definitions.

7.5 Illustrating Example

For the sake of simplicity, let us suppose that there are 5 documents in the knowledge base; we call them LO for "learning object". The corresponding semantic descriptions are shown in figure 7.4. We use the DL sublanguage $\mathcal{EL}$ that has structural subsumption and allows conjunction ($\sqcap$), existential restriction ($\exists r.C$), and the top concept ($\top$). The content of the documents deals with the following topics:

LO_1: information about protocols in general,
LO_2: explanation how the protocol TCP/IP works,
LO_3: explanation that error handling is a task of a protocol,
LO_4: explanation that flow control is a task of a protocol,
LO_5: information about flow control in general.

7.5.1 Step 1: Expanding the Terminology

Expanding the terminology means, making explicit some implicit knowledge. The expanded terminology uses the example taxonomy about networking (see figure 5.1), and is depicted in figure 7.5.

$\mathsf{LO}_1 \equiv \mathsf{Protocol} \sqcap \mathsf{Communication}$
$\mathsf{LO}_2 \equiv \exists \mathsf{howWorks} \sqcap \mathsf{TCPIP} \sqcap \mathsf{Protocol} \sqcap \mathsf{Communication}$
$\mathsf{LO}_3 \equiv \mathsf{Protocol} \sqcap \mathsf{Communication} \sqcap \exists \mathsf{hasTask}.(\mathsf{ErrorHandling} \sqcap \mathsf{ProtocolService} \sqcap \mathsf{Service})$
$\mathsf{LO}_4 \equiv \mathsf{Protocol} \sqcap \mathsf{Communication} \sqcap \exists \mathsf{hasTask}.(\mathsf{FlowControl} \sqcap \mathsf{ProtocolService} \sqcap \mathsf{Service})$
$\mathsf{LO}_5 \equiv \mathsf{FlowControl} \sqcap \mathsf{ProtocolService} \sqcap \mathsf{Service}$

Figure 7.5: Example of an expanded terminology.

7.5.2 Step 2: Computing the Covers

Let us suppose that the user has entered the NL question "What are the tasks of TCP/IP?", and that the question was translated into the following $\mathcal{EL}$-concept description: $Q \equiv \mathsf{TCPIP} \sqcap \exists \mathsf{hasTask}$. In the expanded form the user's question can be denoted as:

$$Q \equiv \mathsf{TCPIP} \sqcap \mathsf{Protocol} \sqcap \mathsf{Communication} \sqcap \ \exists \mathsf{hasTask}.$$

The aim is now to identify the candidate documents within the expanded terminology that cover the expanded query, i.e., that have something in common with Q; these are: LO_1, LO_2, LO_3, and LO_4 as depicted in the following table:

	common with Q	not common with Q
LO_1	$\mathsf{Protocol} \sqcap \mathsf{Communication}$	$\top$
LO_2	$\mathsf{TCPIP} \sqcap \mathsf{Protocol} \sqcap \mathsf{Communication}$	$\exists \mathsf{howWorks}$
LO_3	$\mathsf{Protocol} \sqcap \mathsf{Communication} \sqcap \exists \mathsf{hasTask}$	$\exists \mathsf{hasTask}.(\mathsf{ErrorHandling} \sqcap \mathsf{ProtocolService} \sqcap \mathsf{Service})$
LO_4	$\mathsf{Protocol} \sqcap \mathsf{Communication} \sqcap \exists \mathsf{hasTask}$	$\exists \mathsf{hasTask}.(\mathsf{FlowControl} \sqcap \mathsf{ProtocolService} \sqcap \mathsf{Service})$
LO_5	$\top$	$\mathsf{FlowControl} \sqcap \mathsf{ProtocolService} \sqcap \mathsf{Service}$

7.5.3 Step 3: Computing the Best Cover

Now, for each cover the according Miss and Rest are computed. The best cover is the one with minimal Miss and Rest, with a preference to the minimal Miss as explained in section 7.4.3.

	size of the Miss	size of the Rest
LO_1	$\lvert \mathsf{TCPIP} \sqcap \exists \mathsf{hasTask} \rvert = 3$	$\lvert \top \rvert = 0$
LO_2	$\lvert \exists \mathsf{hasTask} \rvert = 2$	$\lvert \exists \mathsf{howWorks} \rvert = 2$
LO_3	$\lvert \mathsf{TCPIP} \rvert = 1$	$\lvert \mathsf{ErrorHandling} \sqcap \mathsf{ProtocolService} \sqcap \mathsf{Service} \rvert = 3$
LO_4	$\lvert \mathsf{TCPIP} \rvert = 1$	$\lvert \mathsf{FlowControl} \sqcap \mathsf{ProtocolService} \sqcap \mathsf{Service} \rvert = 3$

7.5.4 Conclusion

LO_3 and LO_4 are the best covers and are delivered as an answer to the user's query. Both LOs have the same Miss and Rest, 1 and 3, respectively so that their rank is the same; there is no other LO in $\mathcal{T}$ that has a smaller Miss. It is interesting to mention that the concept TCPIP does not appear in one of the best covers, although it appears in the query and in LO_1. This shows that the best cover is not computed on a statistical evaluation of keywords, but that it is in fact the result of the logical inference.

Other covers, usually those, where the size of the Miss is greater by one than the size of the Miss of the best cover, are yielded as second choice, here: LO_2.

Chapter 8

Benchmark Tests

Contents

In order to evaluate the quality of our E-Librarian Service — with different configurations of the NLP and MIR modules — we run two main benchmark tests. Benchmarks are designed to mimic a particular type of workload on a component or system.

In this chapter, we will report on two benchmark tests. Section 8.1 will give a brief overview of common aspects of the tests, while both benchmarks will be described and evaluated in sections 8.2 and 8.3.

8.1 Overview

In order to evaluate the quality of our E-Librarian Service — with different configurations of the natural language processing (NLP) and multimedia information retrieval (MIR) modules — we run two main benchmark tests. A knowledge base about fractions in mathematics was used for the first test, and a knowledge base about computer networks for the second one. An overview of the NLP and MIR strategies that were employed in the benchmark tests is depicted in figure 8.1.

		NLP strategies (see section 6.3)			
		strategy 1	strategy 2	strategy 3	strategy 4
MIR strategies (see section 7.3)	strategy 1				
	strategy 2		(1)		
	strategy 3				(2)

Figure 8.1: (1) is the first benchmark test over a knowledge base about fractions in mathematics, and (2) is the second benchmark test over a knowledge base about computer networks.

Both benchmark tests were performed on a standard Windows XP computer with a 1.4 GHz CPU and 512 MB of RAM. Our semantic search engines have been implemented as Java applications.

The results achieved with both of our semantic search engines have been compared with the results of a traditional keyword-based search engine. The keyword-based search engine is working in the usual way by browsing the textual content of the documents. The textual content was generated by converting the Powerpoint-slides into pure text. A document is considered as being a potential answer, if at least one (relevant) keyword from the user's query can be found. The keyword-based search engine does not consider stop words, i.e., words with no semantic relevance.

8.2 First Benchmark Test: Fractions in Mathematics

This benchmark test was made in preparation of the planned experiment with the prototype MatES in a school (see chapter 12). The benchmark test confirmed the retrieval quality and reliability of our E-Librarian Service.

8.2.1 Knowledge Base and Set of Questions

The knowledge base about fractions in mathematics is composed of 115 *clips*, which cover all important subjects on fractions as they are taught in secondary schools. A testing set of 229 different questions about fractions in mathematics was created by a mathematic teacher, who was not involved in the development of the prototype. The teacher also indicated the best possible clip manually, as well as a list of further clips that should be yielded as correct answers. The questions were linguistically correct, and short sentences like those that students in a secondary school would ask, e.g., “How can I simplify a fraction?”, “What is the sum of $\frac{2}{3}$ and $\frac{7}{4}$?”, “What are fractions good for?”, or “Who invented the fractions?”.

8.2.2 Evaluation Constraints

Our semantic search engine processed the questions as follows. As shown in figure 8.1, the NLP was based on a part-of-speech tagging of the question, before mapping the tokens to ontology concepts. This strategy is described in detail as "strategy 2" in section 6.3. The MIR was based on standard reasoning services in Description Logics, and the generation of a semantic query (ABox query). This strategy is described as "strategy 2" in section 7.3.

For instance, the semantic interpretation of the question "What is the sum of $\frac{2}{3}$ and $\frac{7}{4}$?" results in the following conjunctive expression: $Fraction(x1) \wedge hasOperation(x1, x2) \wedge Operation(x2, sum)$. For this example, one clip, which explains how to add two fractions, would be retrieved. This would be the best clip that could be found in the knowledge base keeping in mind that our E-Librarian Service returns clips that explain the answer to the student's question, but does not give the precise answer, e.g., it does not compute the sum of the two fractions. This also means that questions like "How can I add two fractions", or "What is $\frac{11}{0.5}$ plus $\frac{5}{5}$" would yield the same clip. On the contrary, the keyword search engine would yield all clips in which keywords like "sum" would be found, e.g., a clip that explains how to represent a complex fraction in terms of additions, and a clip that explains how to describe situations with simple fractions.

8.2.3 Benchmark Results

The processing time of all questions was about 2 – 4 seconds. The results of the benchmark test were the following.

First, the semantic search engine answered 97% of the questions (223 out of 229) correctly, whereas the keyword-based search engine yielded a correct answer (i.e., a pertinent clip) only for 70% of the questions (161 out of 229).

Secondly, the semantic search engine yielded for 86 questions (37%) just one — the semantically best matching — answer (figure 8.2). For 74% of the questions (170 out of 229) the semantic search engine yielded just a few results (one, two or three answers), whereas the keyword-based search yielded for only 14% of the questions less than 4 answers; mostly (138 questions out of 229) more than 10 answers.

Thirdly, our algorithm always returned at least one result. This is important because we know from former experiments that students dislike getting no result at all.

Fourthly, the test also revealed one major weakness of our E-Librarian Service in its current configuration; it is not able to make the difference between a question where there is no answer in the knowledge base, and a question that has no relation to the topic, e.g., "Who invented penicillin?". In other words, the system cannot evaluate the quality of its answer.

8.3 Second Benchmark Test: Computer Networks

This benchmark test was made at the end of the research work to verify the quality of the most advanced NLP and MIR strategies. Although there were fewer clips in the knowledge base used in this test than in the one used in the first benchmark test (see section 8.2), the clips were semantically related, though it was a more complex task for the search engines to find the best clip(s).

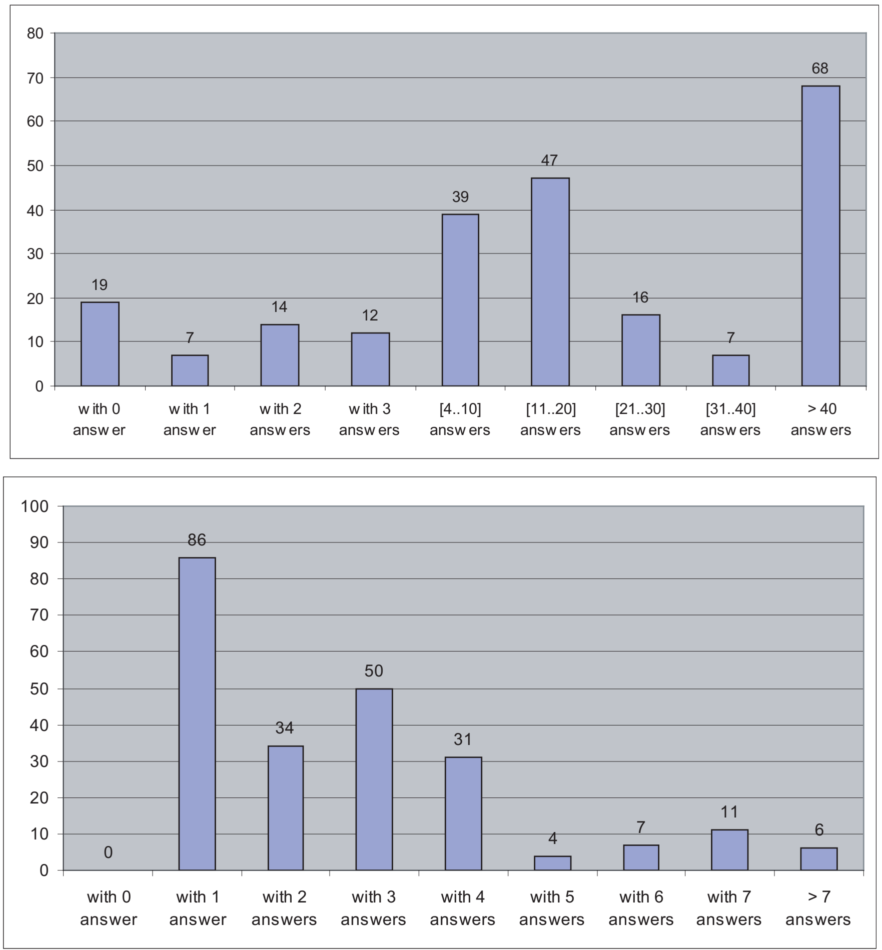


Figure 8.2: Number of results yielded by a (1) keyword-based and by a (2) semantic search engine with a set of 229 questions.

8.3.1 Knowledge Base and Set of Questions

We chose the lecture about Internetworking from the online tele-TASK archive[1] as knowledge base, and split it into 40 smaller clips. A set of 123 user questions about the topic Internetworking has been created. We tried to work out typical student questions, e.g., “What is an IP-address composed of?”, “How does a datapacket find its way through a network?”, “What is a switch good for?”, or “Do internetprotocols guarantee an error-free communication?”. For each question, we also indicated the relevant answer(s) that should be supplied.

8.3.2 Evaluation Constraints

As shown in figure 8.1, the NLP was based on a partial parsing (chunking) of the user question, before mapping the tokens to ontology concepts. This strategy is described in detail as “strategy 4” in section 6.3. The MIR was based on the computation of *best covers* in Description Logics. This strategy is described as “strategy 3” in section 7.3.

[1] `http://www.tele-task.de/page42_mode1_series599.html`

For example, the question "What is an IP-address composed of?" would be translated into the $\mathcal{EL}$-concept description: $\mathsf{IPAddress} \sqcap \exists \mathsf{isComposedOf}$. Then, the semantic distance between this query and the semantic description of the clips in the knowledge base is computed in terms of Miss and Rest. Only the clips with the highest similarity would be yielded as an answer to the user question.

We call an answer from our semantic search engine a *perfect hit* if it covers the query completely, i.e., where Miss = Rest = 0. We call an answer a *sufficient hit* if it covers the query completely, but the answer contains more information than necessary, i.e., where Miss = 0 and Rest > 0.

For the evaluation, we only considered the best covers with minimal Miss, not the second choices. This means that if the E-Librarian Service did not deliver an exhaustive answer (Miss = 0) as best cover but only as second choice, then we considered the answer to be wrong. This constraint was not given in the first benchmark test (see section 8.2).

8.3.3 Benchmark Results

The processing time of the first question is about 200 ms, while for the rest it is less than 10 ms. The outcomes of the benchmark test are the following.

First, our semantic search engine scored better than the keyword-based search regarding the pertinence of the results. In most cases the E-Librarian Service yielded the correct answer as depicted in the following table:

	perfect hits	sufficient hits	total queries
E-Librarian Service	93 (76%)	112 (91%)	123 (100%)
Keyword search	9 (7%)	103 (84%)	123 (100%)

These numbers emphasize the pertinence of our E-Librarian Service as an appropriated tool for an educational environment; in most cases the learner gets a satisfying, even perfect, answer from the system. The fact that some answers contain little more information than necessary is no problem, and can even have a positive effect on the learner.

Secondly, the precision of our retrieval algorithm is confirmed by the fact that on average less than 0.7 clips are returned in addition to the perfect answer (compared to 6 clips for the keyword-based search). Figure 8.3 shows the number of supplementary clips being supplied in addition to the expected answer. This important outcome points out that the E-Librarian Service usually achieves the correct answer with no additional information (93 out of 123), and in a few cases one (12 out of 123) or two (6 out of 123) supplementary clips. This also emphasizes the major improvement of this MIR strategy compared to the first benchmark test (see section 8.2); here, a much higher precision is achieved. The keyword-based search engine in general returns a lot more secondary clips.

This result is an important evidence for the pertinence of our E-Librarian Service in an educational environment; the user asks a precise question (or enters a keyword phrase) and expects few but concise answers. However, the keyword-based search leaves the user with the awkward task of filtering the pertinent answers out of the noise.

Thirdly, in information retrieval the performance of a retrieval algorithm is measured by *recall* and *precision* [BYRN99]. The recall is the fraction of the relevant documents (R) that has been retrieved, and precision is the fraction of the retrieved documents (A) that is relevant, written:

$$Recall = \frac{|Ra|}{|R|} \qquad Precision = \frac{|Ra|}{|A|}$$

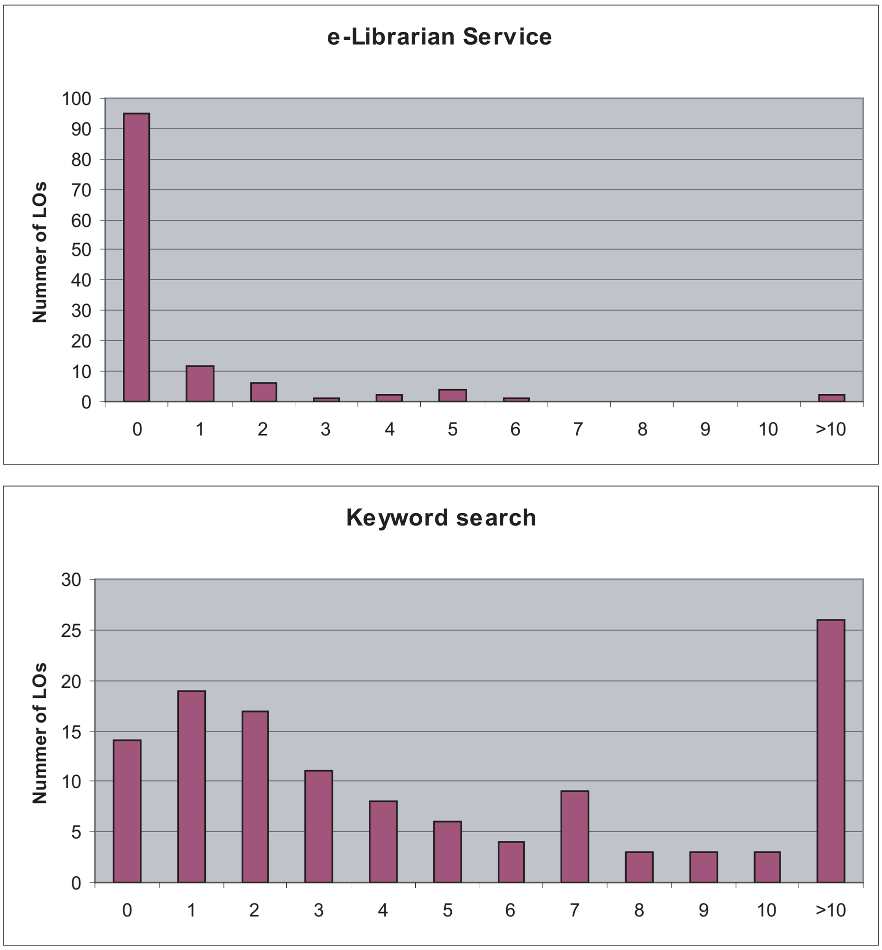


Figure 8.3: Number of supplementary clips yielded with the optimal answer.

where $|Ra|$ is the number of retrieved and relevant documents so that $Ra = R \cap A$. For each query, we computed its precision versus recall curve. To properly evaluate and compare the quality of both retrieval algorithms, we averaged the precision figures at each recall level as follows:

$$\overline{P}(r) = \sum_{i=1}^{N_q} \frac{P_i(r)}{N_q}$$

where $\overline{P}(r)$ is the average precision at the recall level r, N_q is the number of queries used, and $P_i(r)$ is the precision at recall level r for the i-th query.

Our curves are based on the 11 standard recall levels, which are 0, 10, 20,..., 100. Since the recall levels for each query might be distinct from the 11 standard recall levels, utilization of an interpolation procedure was necessary. The interpolation used is the following. Let $r_j, j \in \{0, 1, 2, ..., 10\}$ be a reference to the j-th standard recall level (i.e., r_5 is a reference to the recall level 5), then: $P(r_j) = max_{r_j \leq r \leq r_{j+1}} P(r)$.

Figure 8.4 shows the average precision curve for both retrieving algorithms: the E-Librarian Service, and the keyword-based search. Generally, the precision curves fall with increasing recall. This is not the case in our evaluation due to the fact that for each question in the test set, there

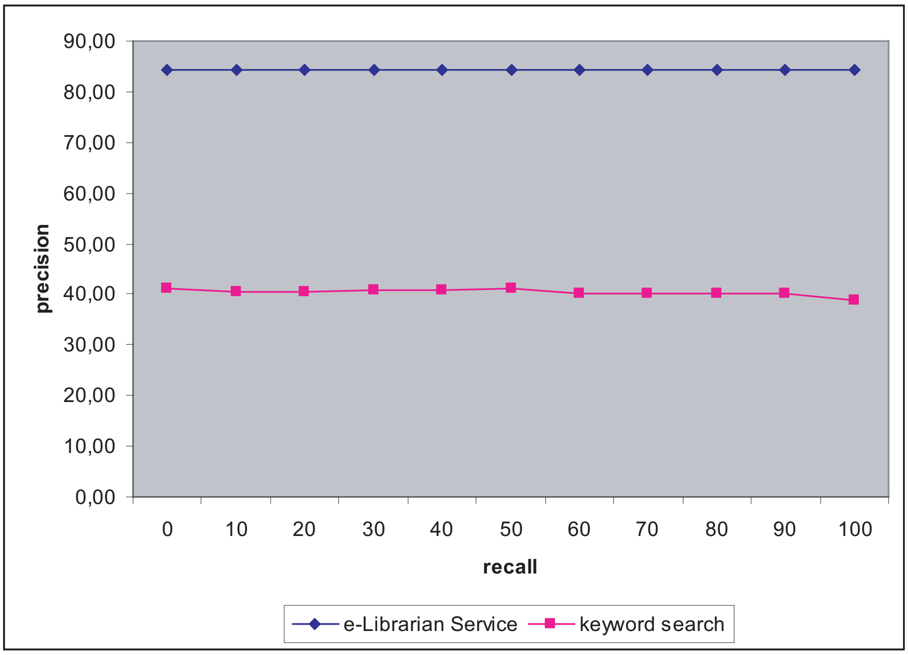


Figure 8.4: Average precision at 11 standard recall levels.

are only few relevant documents to be retrieved (in average 1.29 relevant answers per question). For an average recall-level, the precision of the algorithm is 84.41%, compared to 40.42% for the keyword-based search.

These numbers confirm the previous outcome that our algorithm has a very high precision concerning the pertinence of the yielded answers; its average precision is more than twice as much than the precision achieved with the keyword-based search.

Chapter 9

Implementation and Prototypes

Contents

Our background theory for an E-Librarian Service was implemented prototypically in three different educational tools. Section 9.1 will give an overview of the different prototypes. The layer architecture of our E-Librarian Service will be depicted in section 9.2. Technical details about the development of the prototypes will be described in section 9.3.

9.1 Prototypes

During this research project, we developed and tested different prototypes: CHESt with a knowledge base about computer history, MatES with a knowledge base about fractions in mathematics, and the most advanced prototype called "The Lecture Butler's E-Librarian Service" with a knowledge base about computer networks and Internetworking. All prototypes will be described in this section. Figure 9.1 gives an overview of the different natural language processing (NLP) and multimedia information retrieval (MIR) strategies implemented in each prototype.

		NLP strategies (see section 6.3)			
		strategy 1	strategy 2	strategy 3	strategy 4
MIR strategies (see section 7.3)	strategy 1	CHESt v2			
	strategy 2		CHESt v3 MatES	CHESt v4	
	strategy 3				Lecture Butler

Figure 9.1: Overview of the NLP and MIR strategies implemented in the different prototypes.

9.1.1 Computer History Expert System (CHESt)

A first prototype of our E-Librarian Service is called CHESt, which stands for *Computer History Expert System*. It was firstly published in [LM04a]. The tool has a knowledge base with 300 multimedia clips in German language that cover the main events in computer history. The author of this thesis recorded the clips at the University of Trier in the summer of 2003 using tele-TASK[1] [SM02]. Because CHESt was the first prototype, it is the one that changed most with the improvement of the NLP and MIR strategies. In its final version, CHESt v4, it implemented "strategy 3" for its NLP, and "strategy 2" for the MIR. *Pellet*[2] [SP04] was used as external reasoner.

The MIR module was implemented in Java. The user can access the search engine via a Windows application (see figure 9.2) developed in Delphi[3]. CHESt is completely distributed; the semantic search engine, the knowledge base, and the user application can be on different machines. The communication is transparent and is done using socket connections.

CHESt was used in different experiments in a school (see chapter 11). For this, different configurations were elaborated; it turned out that the following two were the most suitable:

- The search engine was installed on a server in the LAN, and all users had a local copy of the client application. The knowledge base was accessed via a streaming server. This configuration is the most secure one, because nobody has direct access to the clips and to the search engine. Unfortunately, the quality and performance of the network play a key role in the reliability of this configuration.
- All components are on the user's machine. The complete prototype — knowledge base, search engine, and client application — fit on one CD-ROM. Although this configuration has the highest reliability and performance, it is the most unsafe; everyone can make a copy.

[1] `http://www.tele-task.de`
[2] `http://www.mindswap.org/2003/pellet/`
[3] http://www.codegear.com/products/delphi

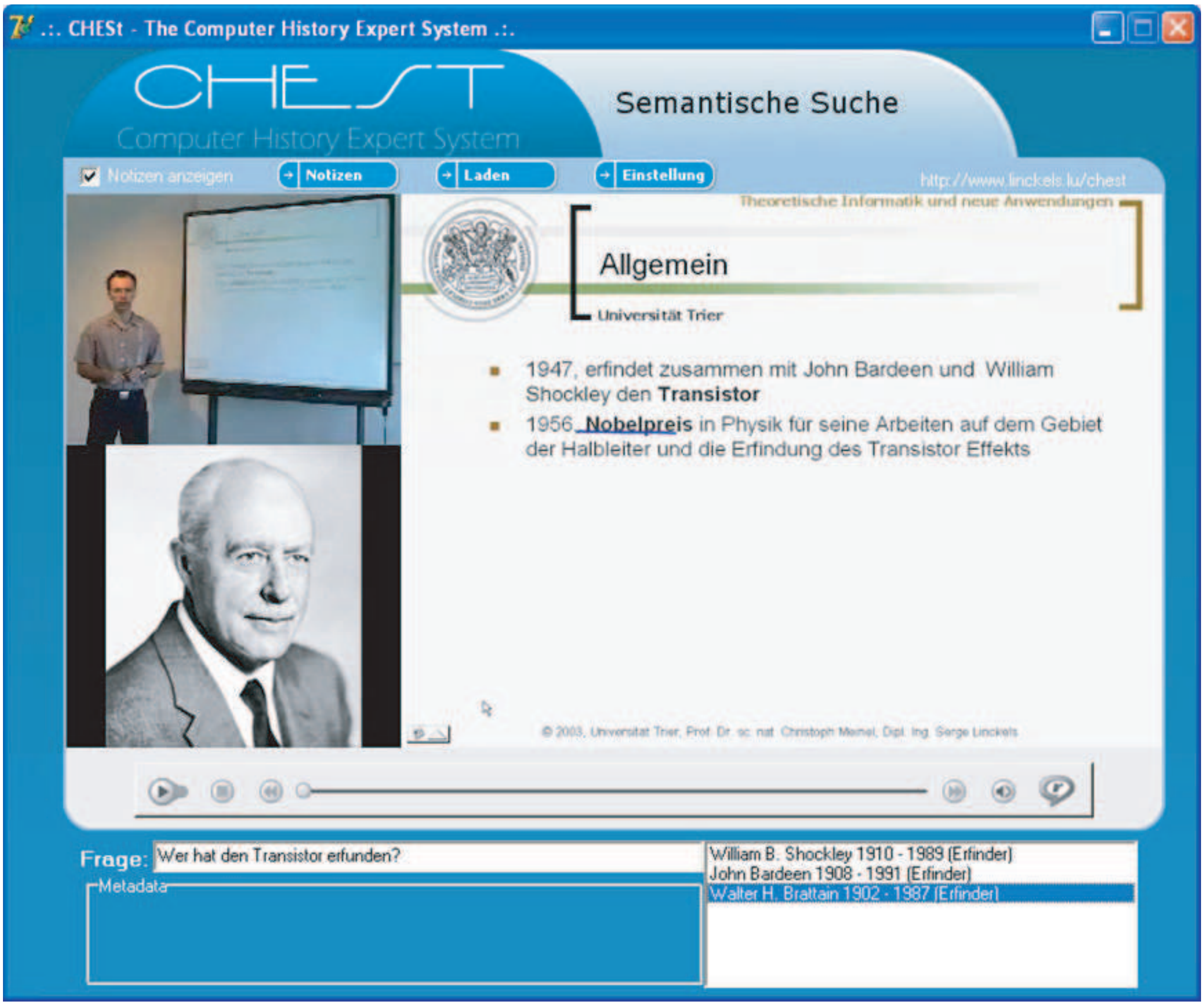


Figure 9.2: CHESt with a semantic search and the question: "Who invented the transistor?".

A useful feature of the prototype CHESt is that the user can take notes. While watching a clip, the user can at any time pause the playback, and add a note at that time position. Next time the clip is played, this note is displayed at the same time-mark. In the current state of this prototype all notes are stored locally.

The prototype CHESt was developed from summer 2003 to summer 2005. It was no longer maintained when the development of a new prototype, MatES, began in the summer of 2005.

9.1.2 Mathematics Expert System (MatES)

A second prototype of our E-Librarian Service is MatES that stands for *Mathematics Expert System*. It was firstly published in [LM06b]. MatES was specially design to be used for a pedagogical experiment in a secondary school in Luxembourg (see chapter 12). The knowledge base covers the topic of fractions in mathematics w.r.t. the official school programme. Most of the 115 clips were recorded by pupils at the Lycée Technique Esch/Alzette (LTE) using tele-TASK. The slides were produced by Carole Dording as a part of her teacher training. The clips were recorded in French because it is the language used to teach mathematics in Luxembourg. Therefore, the search engine

Figure 9.3: MatES with the question: "How to divide one fraction by another fraction?".

used a French dictionary. MatES has the same architecture as CHESt in its final version.

Being identical to CHESt, the semantic search engine was implemented in Java. The user application (see figure 9.3) was developed in Delphi. Since MatES was created for students to be used at home or in a classroom, it was mainly used in the following two configurations:

- A "Home Edition", where the knowledge base, the semantic search engine, and the user application are on one DVD. No configuration or installation procedure is necessary to use MatES. This configuration was used in the experiment because it is the most reliable and the simplest to use.

- A "Classroom Edition", where the semantic search engine and the knowledge base are installed on an application server in the LAN, and every student has a local copy of the client application. This configuration is more secure than the first one, because nobody has direct access to the clips and the search engine. Unfortunately, the quality and performance of the network play a key role in the reliability of this configuration.

In both cases, no streaming server is required, and the user has a very high speed access to the clips. The response time of the system to process a question and to deliver its answer is between 3 and 5 seconds.

The prototype MatES was developed from summer 2005 to winter 2006 before being used in a pedagogical experiment (see chapter 12) in summer 2006. It was no longer maintained after that experiment because we started with the development of a better prototype: "The Lecture Butler's E-Librarian Service".

9.1.3 The Lecture Butler's E-Librarian Service

The most recent and advanced prototype of our E-Librarian Service has been developed in the context of the Web University project at the HPI[4], which aims at exploring novel Internet- and IT-technologies in order to enhance university teaching and research. The objective is to create a tool for the learner — we call it the "Lecture Butler" — that assists the student in his/her learning process. The Lecture Butler is a collection of different utilities, e.g., for creating a personalized lecture flow [KLM07], for mobile learning [WLM07], or for the semantic indexing of lecture videos [RM06]. Another feature of the Lecture Butler is our E-Librarian Service.

The development of the Lecture Butler's E-Librarian Service started in summer 2006, and was firstly mentioned in [LRKM07]. It focuses on a lecture series from the online tele-TASK archive about Internetworking ("Internet- und WWW-Technologien") by Prof. Christoph Meinel at the HPI[5], which is a set of 30 units with a total of 38 hours of recorded lectures. We split the lecture units into smaller clips, with the idea that users generally ask short and precise questions, and expect short and precise answers. They prefer short clips with a length of some minutes instead of complete lectures of 90 minutes. This topic is discussed in more detail in chapter 10.

Contrary to the former prototypes, the "Lecture Butler's E-Librarain Service" is a Web application. It offers a standardized Web interface to respect a service oriented architecture (SOA). This allows potentially every developer to write his/her own application that uses the search capabilities of our E-Librarian Service. This feature is described in more detail in section 9.3.3. We implemented a Web site as client interface for the prototype (see figure 9.4).

9.2 Architecture

Our E-Librarian Service is an interactive and distributed system. The principle is to keep the required technology at the user-side as simple as possible, so that no special installation or configuration is necessary. Inspired by the Semantic Web technologies and theories (see chapter 2), we developed our E-Librarian Service around a straight-forward layer architecture. In general, our architecture is composed of four main layers (see figure 9.5): the Knowledge Layer, the Inference Layer, the Communication Layer, and the Presentation Layer. Throughout the whole project, this modularized approach was helpful in the extension and evolution of the E-Librarian Service; modules could be exchanged by new ones without affecting the complete system. For example, when the MIR module was improved by a new algorithm, we simply exchanged that module, and the NLP module or the communication module were not affected.

In this section, we will give an overview of the architecture, and will describe the main functions of each layer. The architecture was first published in [LM04d].

[4] http://www.hpi.uni-potsdam.de/~meinel/research/web_university.html
[5] http://www.tele-task.de/page42_mode1_series599.html

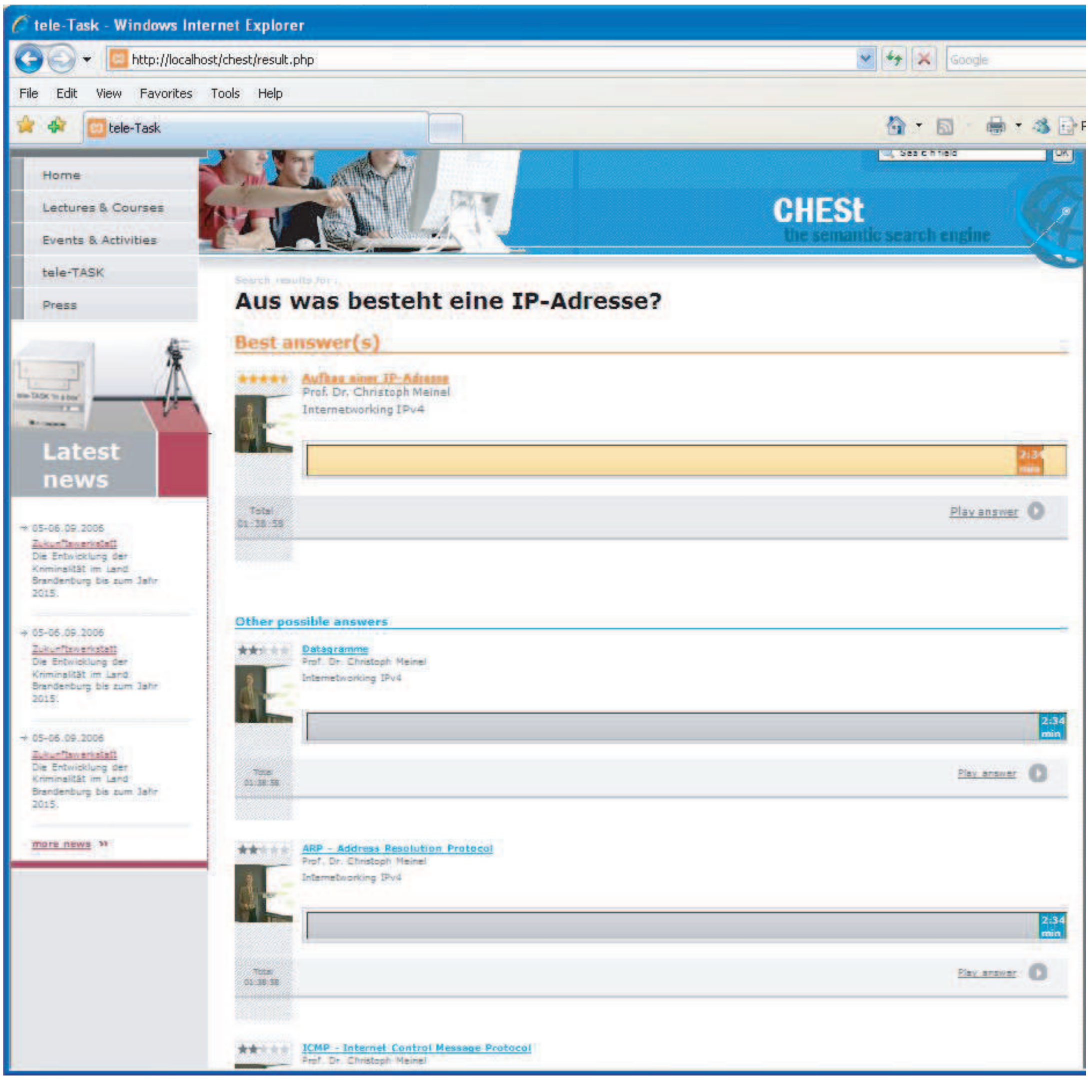


Figure 9.4: Presentation of the search results with the Lecture Butler's E-Librarian Service for the question: "What is an IP-address composed of?".

9.2.1 Knowledge Layer

The Knowledge Layer is the set of data sources which are accessed by the Inference Layer for reasoning over the knowledge. The data sources are: the domain dictionary, the domain ontology, and a terminology. The domain language is used for the NLP (see section 6.4). The domain ontology gives the necessary information about the complete domain (see section 5.2). The terminology delivers the semantic description of the documents in the knowledge base (see section 7.4). Both, the ontology and the terminology, are encoded as OWL-files.

9.2.2 Inference Layer

The Inference Layer is the most important one because it implements the semantic search engine; all the reasoning is done at this level. It is composed of the NLP module (see section 6.4), the MIR module (see section 7.4), and the different interface sub-layers (see sections 9.3.2 and 9.3.3). In

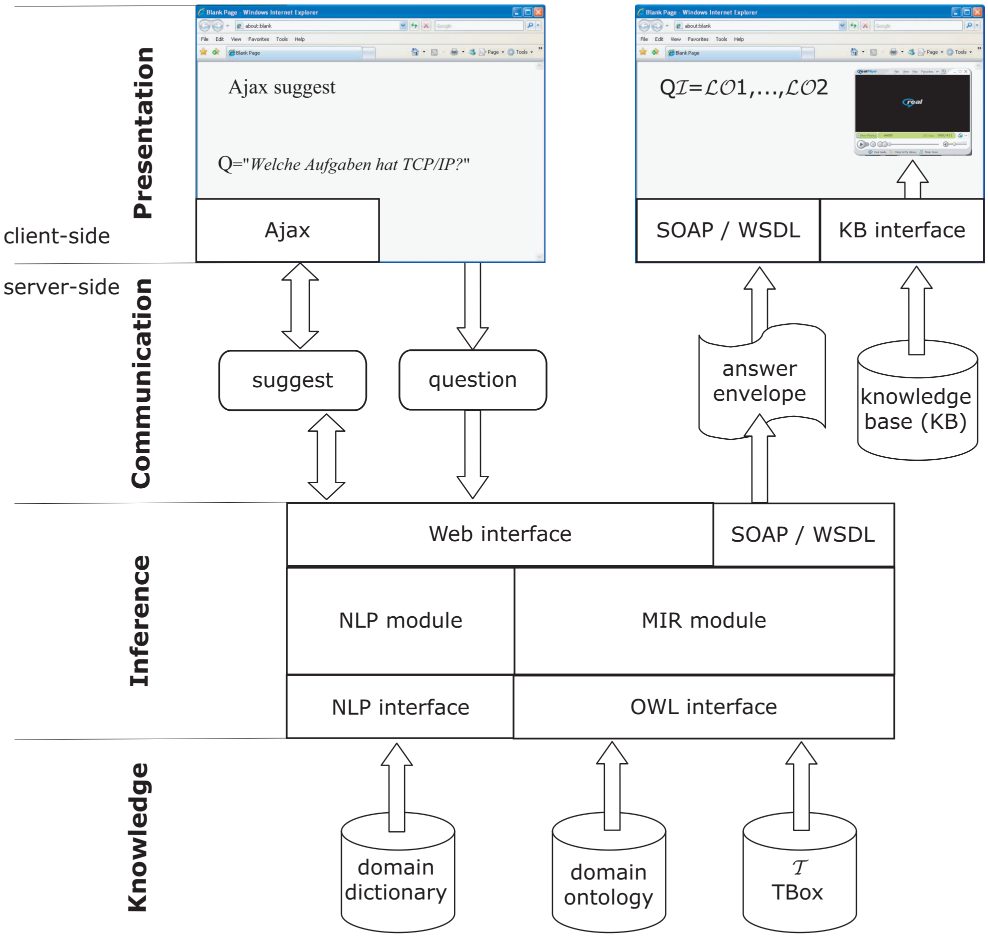


Figure 9.5: General architecture of our E-Librarian Service.

a simplified view, the Inference Layer works as follows: it receives a question in natural language (NL) from the upper layer, translates it into a logical form, computes the best answer, and returns the references (URIs) of the pertinent documents to the upper layer. In addition, the inference engine is able to "understand" if it has the correct — or at least a good — answer, and is able to explain to the user the quality of the delivered answer, i.e., the semantic distance between the entered question and the yielded documents.

The communication with the upper Presentation Layer is based on two requirements. First, the answer to the user's question must be available for the user in a very short time. This excludes sending the whole resulting clip(s) to the user. Instead, only the URI — where the clip(s) can be retrieved — and some metadata about the document(s) are transmitted. It is up to the client application how this data will be presented to the user. Secondly, the Inference Layer must be transparent to the user, independently of the fact whether the inference engine runs as a process (service) on the user's local machine, or if it is accessed distantly as a Web service. Hence, the answer of the Inference Layer must be encoded in a platform and system independent way. In our most advanced prototype, the list of pertinent documents is encapsulated into a *SOAP envelope* (see section 9.3.3) that can be unfolded on the client side.

9.2.3 Communication Layer

The Communication Layer allows a transparent communication between the client application (Presentation Layer) and the semantic search engine (Inference Layer). It should not be important if these two layers are on the same machine or not. Furthermore, the communication must be error-free, transparent and hardware independent.

In the most advanced prototype of our E-Librarian Service, we employed a service oriented architecture (SOA) that offers a standardized access to the inference engine via a SOAP/WSDL interface (see section 9.3.3). This allows developers to create their own client application that communicates with the inference engine, and is fully compatible with the current evolution of Semantic Web [BBZ+05].

In other prototypes, the Communication Layer was based on a low level socket communication. Using sockets has three advantages: it uses the popular protocol TCP/IP, it offers an error-free transmission, and components for most development environments are widely available.

9.2.4 Presentation Layer

The Presentation Layer represents the interface between the user and the machine. It gets a question from the user and transmits it to the semantic search engine via the Communication Layer. In return, it displays the search result(s), and allows the user to watch the clip(s). In our most advanced prototype, the Presentation Layer is implemented as a Web page (see section 9.3.2). Pedagogical and ergonomic aspects of the human-machine interface are discussed in chapter 10.

We would like to mention different possible improvements. First, an interface that adapts automatically to the user: simple interface for kids, more expressive interface for experts. Beside the pure layout problems, XSLT [HM02] could be used to filter too complicated documents from the resulting set. This is due to the fact that different users prefer different levels of complexity, of difficulty, and of elaborateness. Other interesting approaches are, e.g., scalable metadata [LTV03], or meta-modeling teachware [SF99].

9.3 Development Details

In this section we will describe three technical details about the implementation of our E-Librarian Service: the processing of OWL and Description Logics (DLs) in Java, a special "auto-suggest" feature for the input of NL, and the Web service interface.

9.3.1 Processing OWL and DLs in Java

The semantic search engine was developed in Java in order to guarantee platform independency. The *Jena API*[6] [CDD+04] is most commonly used in Java to process RDF/OWL. However, Jena has very limited reasoning capacities; let us remember that our most advanced MIR strategy is mainly based on non-standard inferences like the concept covering problem (see section 7.4). Furthermore, as we deal with a large number of concepts and a rather complex ontology, we have found out that Jena slowed down the performance of our prototype. Therefore, we rely only on Jena for some basic operations like loading the terminology and the ontology.

Instead, we developed our own optimized datastructure for representing DL-concept descriptions, and algorithms for reasoning over the data (see figure 9.6). In our vocabulary, a DL-concept

[6] `http://jena.sourceforge.net/`

description is composed of generic terms (`DLTerm`). Each term is identified by its URI. For example, the concept TCPIP has the following URI:

`http://www.linckels.lu/chest/WWW/elements/1.0#TCPIP`

The classes `DLConcept` (lines 8 – 12) and `DLrole` (lines 14 – 19) inherit from the generic `DLTerm` (lines 1 – 6), and represent DL-concepts and DL-roles respectively. `DLRole` is special because a role can have a list of arguments (line 15), which again are of the type `DLTerm`. Finally, a DL-concept description — implemented in the class `DLDescription` (starting at line 26) — is, among a set of operations, simply a list of terms (line 28).

```
1    class DLTerm {
2   // Generic class(concept or role)
3     String uri;
4     DLTerm(String uri) {
5       this.uri = uri;
6     } // constructor DLTerm
7   } // DLTerm
8
9   class DLConcept extends DLTerm {                     // A DL concept
10    DLConcept(String uri) {
11      super(uri);
12    } // constructor DLConcept
13  } // DLConcept
14
15  class DLRole extends DLTerm {                           // A DL role
16    DLDescription args;
17    DLRole(String uri) {
18      super(uri);
19      args = new DLDescription();
20    } // constructor DLRole
21
22    void add(DLTerm t) {
23      // adds a term to the arguments
24      args.add(t);
25    } // add
26  } // DLRole
27
28  class DLDescription {                    // A DL-concept description
29    ArrayList terms;
30    DLDescription() {
31      super();
32      terms = new ArrayList();
33    } // constructor DLDescription
34    ...
35  }
```

Figure 9.6: Our datastructure for representing DL-concepts, -roles and -concept descriptions.

We explored different Java containers before deciding to use *ArrayList*[7]. This special type of container is a generic and resizable array of objects, which comes with highly optimized operations. With Jena, a set of 120 queries was processed in approximatively 5 minutes, whereas the same set of questions was processed in approximatively 10 seconds with our datastructure and algorithms.

For illustration, we describe two operations over our datastructure: quantification of the size of a DL-concept description, and the computation of the least common subsumer.

[7] `http://java.sun.com/j2se/1.4.2/docs/api/java/util/ArrayList.html`

9.3.1.1 Quantification of the Size of a DL-Concept Description

We developed the method `quantify()` for the class `DLDescription` that quantifies the size of a DL-concept description according to definition 23 (see section 7.4.2). It is invoked with the command `C.quantify()`, where `C` is an instance of the class `DLDescription`. The result of the method is a positive integer. The code is shown in figure 9.7.

It works as follows. A standard iterator[8] is used to browse through the list of terms in the DL-concept description (line 3). Each term adds at least the value 1 to the size n of the DL-concept description (line 5). If the term is a role, then n is incremented by the size of the role's arguments, which are recursively computed (line 7).

```
1    int quantify() {
2      int n = 0;
3      for (Iterator i = terms.iterator(); i.hasNext();) {
4        DLTerm t = (DLTerm) i.next();
5        n++;
6        if (t instanceof DLRole)
7          n = n + ((DLRole) t).args.quantify();
8      } // for i
9      return n;
10   } // quantify
```

Figure 9.7: Method that quantifies the size of a DL-concept description.

9.3.1.2 Computation of the Least Common Subsumer

We developed the method `lcs()` for the class `DLDescription` that computes the least common subsumer (lcs) of two DL-concept descriptions according to definition 2 (see section 3.2.2.2). It is invoked with the command `D.lcs(C)`, where `C` and `D` are instances of the class `DLDescription`. The result of the method is a DL-concept description representing the lcs of `C` and `D`. The code is shown in figure 9.8.

It works as follows. A standard iterator is used to browse through the terms of the DL-concept description `D` (line 5). It is tested for each term if it also exists in the DL-concept description `C` (lines 9 and 10). The method `contains` returns *true* if a given term is found in the referenced DL-concept description. If the term exists in `C`, then it is added to the resulting list of the lcs if it is a concept (line 26 – 27). If the referenced term is a role (line 13), then both roles must be compared. There are three possibilities: the role-name and the role-arguments match, then the complete role is added to the resulting lcs-concept description (lines 17 – 19), only the role-names match, then only the role-name is added to the resulting lcs-concept description (lines 20 – 22), or there is no match at all, then the role is ignored.

9.3.2 Client Front-End with Ajax Autocompleter

Let us put into evidence that the basic task of the Presentation Layer is to allow people — mostly not computer experts — to express their questions in NL, and to watch the resulting document(s). Therefore, the graphical user interface (GUI) must be as simple and ergonomic as possible.

A feature of our most advanced prototype is that it helps the user to compose his/her NL question; we used *Ajax* [Eer06] to create a "suggestion textfield" (or autocompleter). When the user strikes a key, then this information is immediately transmitted to the Web server, which returns

[8] `http://java.sun.com/j2se/1.4.2/docs/api/java/util/Iterator.html`

```
DLDescription lcs(DLDescription C) {
  DLDescription res = new DLDescription();

  // Browse through description of this object
  for (Iterator i = terms.iterator(); i.hasNext();) {
    DLTerm t = (DLTerm) i.next();

    // check if this term is in C
    DLTerm u = C.contains(t);
    if (u != null) {

      // t is a role
      if (t instanceof DLRole) {
        DLRole r = (DLRole) t;

        // check if the roles' arguments match
        if (r.args.equal(((DLRole) u).args))
          // yes -> add the complete role
          res.add(r);
        else
          // no -> add only name of role (must be new role)
          res.add(new DLRole(r.uri));
      } // if role

      // t is a concept
      else
        res.add(t);
    } // if not contained
  } // while

  return res;
} // lcs
```

Figure 9.8: Method that computes the lcs.

a list of words that start with this character. The user can select one word of that suggestion-list, or can continue to type the word. For each additional character, the suggestion list is refreshed.

This feature has at least three advantages. First, it helps the user to quickly assemble the words to form a complete sentence. Secondly, the risk of spelling errors in the sentence is reduced. Thirdly, only words from the system's domain dictionary are used, which will result in a very reliable semantic interpretation of the user's questions. Of course, the user can use words that are not known by the system. In that case, these words are logged and can be added later to the domain dictionary by the administrator. In that case however, these words will be ignored in the semantic interpretation.

The code of the suggestion textfield is shown in figure 9.9. The *autocompleter* is initialized when the Web site is loaded (line 7). This means that an instance of the autocompleter-class `MyAutocompleter` is created, and a reference to the textfield `autocomplete` and the layer `autocomplete_choices` is made (lines 2 – 3). The variable `suggestions` is a JavaScript-array that contains the list of all possible words, i.e., a local copy of the domain dictionary. The suggestion textfield is a standard HTML-textfield that references the Ajax-class `autocomplete` (line 9). The HTML-layer `autocomplete_choices` is the window that contains the list of suggested words (line 10). It is automatically refreshed each time the user strikes a key.

```
<html> <head> ...
 function initAutocompleter() {
   new MyAutocompleter( "autocomplete","autocomplete_choices",suggestions );
 }
...
</head>
<body onLoad="initAutocompleter()">
   ...
   <input type="text" id="autocomplete" name="txtQuestion" size="50" />
   <div id="autocomplete_choices" class="autocomplete"></div>
   ...
</body> </html>
```

Figure 9.9: Textfield with Ajax suggestion list.

9.3.3 The SOAP Web Service Interface

In a former version of our E-Librarian Service, we used sockets for the communication between the Presentation Layer and the Inference Layer. This low-level communication was reliable and sufficient. But, to offer a more standardized and modern communication interface to our E-Librarian Service, we explored different alternative solutions, e.g., the Grails framework[9]. Finally, we decided to implement our E-Librarian Service as a Web service with a standardized SOAP interface. This has the advantage that every developer can build his/her own application that uses our semantic search engine, while the transportation of the NL question and the set of yielded documents remains transparent.

In our prototypical implementation, we use the open source *Apache Axis*[10] [WBFT04] as Web service framework. It consists of a Java and a C++ implementation of the SOAP server, as well as various utilities and APIs for generating and deploying Web service applications. When a Web service is exposed using Axis, it will generate a *WSDL* file automatically when accessing the Web service URL with `?WSDL` added to it. The *Web Services Description Language* (WSDL) is an XML-based language that provides a model for describing Web services. Version 2.0 is a W3C recommendation[11]. The WSDL defines services as collections of network endpoints, or ports. Finally, the complete E-Librarian Service can easily be assembled in an .aar-file — which is nothing else than a Java archive (jar) — and deployed as a Web service. It is accessible via the HTTP interface, e.g., `http://theHostName:8080/axis2/services/CHESt?wsdl`.

SOAP is a protocol for exchanging XML-based messages over computer networks, normally using HTTP(S). SOAP originally stood for *Simple Object Access Protocol*, and lately also S*ervice Oriented Architecture Protocol*, but is now simply SOAP. The original acronym was dropped with Version 1.2 of the standard, which became a W3C Recommendation[12] in June 2003, as it was considered to be misleading. There are several different types of messaging patterns in SOAP, but by far the most common is the *Remote Procedure Call* (RPC) pattern, in which one network node (the client) sends a request message to another node (the server), and the server immediately sends a response message to the client. SOAP makes use of an Internet application layer protocol as a transport protocol. Both SMTP and HTTP are valid application layer protocols used as transport for SOAP, but HTTP has gained wider acceptance as it works well with today's Internet infrastructure. XML was chosen as the standard message format because of its widespread use

[9] `http://grails.codehaus.org/`
[10] `http://ws.apache.org/axis/`
[11] `http://www.w3.org/TR/wsdl`
[12] `http://www.w3.org/TR/soap/`

by major corporations and open source development efforts. Additionally, a wide variety of freely available tools significantly ease the transition to a SOAP-based implementation.

```
1   ...
2   // read question from HTML-form
3   $question = $_POST['txtQuestion'];
4
5   // submit query to Web service via SOAP
6   require_once 'SOAP/Client.php';
7   $wsdl_url = 'http://localhost:8080/axis2/services/CHESt?wsdl';
8   $WSDL = new SOAP_WSDL($wsdl_url);
9   $client = $WSDL->getProxy();
10  $params = array("$question");
11  $answer = $client->search($params);
12  ...
```

Figure 9.10: SOAP communication with the semantic search engine.

Figure 9.10 illustrates how our E-Librarian Service can be accessed via SOAP. The example shows some PHP-code that reads a user question (line 3). The connection with the Web service is established (lines 6 – 9), and the remote procedure `search()` is called with the user question as argument (lines 10 – 11). The remote procedure returns the answer encoded as string. In our prototype, this answer-string contains the URIs of the different documents, as well as ranking information, i.e., the semantic distance.

Part III

Applications

Chapter 10

Pedagogical Aspects

Contents

E-learning is a general term used to refer to computer-enhanced learning. Although E-Learning is today well established in educational communities, its dramatically announced benefits have not been proven yet.

This chapter will start with a general and critical introduction to E-Learning in section 10.1. Then, we will describe the pedagogical advantages of our E-Librarian Service in section 10.2.

10.1 E-Learning — A Critical View

The buzzword E-Learning was introduced in the 1980s as a promise that the use of computers in schools will improve learning. Although E-Learning is today well established in educational communities, its dramatically announced benefits have not been proven yet. This section will give a more critical view of E-Learning. We will start with a short historical introduction, before presenting recent pedagogical E-Learning practices. Finally, we will report on surveys about how teachers and parents perceive new technologies. Most of this work was published in [LM04c, LME05, LM06a, LKM06].

10.1.1 Promises or Reality?

Once upon a time, some people dreamed about fundamentally changing the way of teaching by replacing instructors by television sets. Initially hopes were high that television would have certain characteristics that would lead to improved student learning, but none have been found. A new vision of the way people learn was triggered by the birth of personal computers in the late 1970s and beginning 80s. The maturity of computer hardware, especially in size and price, was for artificial intelligence (AI) scientists the long awaited event to invade the world of education. But instead of smart AI tools, the software industry inundated the world market with "educational software". The tremendous impact of the Web in the mid 90s was perceived as a new change of the educational paradigm. The richness of the Web promised to make home schooling an increasingly popular option for parents. Foshay, Silber & Stelnicki state in their handbook [FSS03] that most learning takes place in the natural work environment through social interaction (coaching) and team collaboration rather than in the classroom. Also, in the recent years, the advances in multimedia technologies and the raising availability of broadband access to the Internet are the base for creating new educational technologies like "online education", "collaborative learning", "Virtual Reality Learning" [YH03], or "E-Learning 2.0". The term E-Learning 2.0 has been used to refer to the use of social software such as blogs and wikis [Dow05]. We have given a more detailed view about modern E-Learning techniques in [LME05, LKM06].

Today, modern schools are built with "cyber-age technologies" in mind. But although the features of E-Learning sound very promising, the real gain compared to traditional courses has not yet been proved. Some statements at "didacta 2005" — the trade fair for education and training — were that "several models did not work", and that "there is a lot of scrap" among the huge heap of educational software[1].

Owstens [Ows97] says that we cannot simply ask: "Do students learn better with E-Learning technologies as compared to traditional classroom instruction?". The key to improve learning with new technologies appears to depend on how effectively the particular technology is exploited in the teaching-learning situation [TFG05]. In his survey about the benefits of E-Learning technologies Bonk [Bon01] publishes that nearly 40% of the teachers reported that they were unsure, while 32% noted that course quality was in fact improved, and another 29% said that it was not. Burdman [Bur98] refers to a study from Rensselaer Polytechnic Institute in New York, which reports that some students do a little better, some students do a little worse, and for the rest there is no significant difference.

Warschauer [War06a] concludes that there had not been any appreciable effect on student test scores, but working with multimedia on a daily basis in school creates higher levels of student engagement, and involved students spend more time on tasks, work more independently, enjoy learning more, and take part in a greater variety of learning activities at school and at home.

[1] Spiegel online: `http://www.spiegel.de/unispiegel/studium/0,1518,344754,00.html`

Sloan and Borse [PBR06] found out that students were more motivated when they used tablet PCs and pen-based technology. Warschauer [War06b] states that improved integration of technology in schooling is obviously a key element for overcoming the strong home-school disconnection in how children learn.

10.1.2 Exploratory Learning and Edutainment

Current educational thinking is that students are better able to master, retain, and generalize new knowledge when they are actively involved in constructing that knowledge in a learning-by-doing situation [You98]. This leads to the expression of *exploratory learning*, which usually emphasizes using computers as tools rather than as teachers. New attention has been given to teaching methods — the pedagogy — in a computer-based learning environment [Dit03], e.g., in a Web-based education [Bau96, BB04, HMS05].

Students tend to be more *visual learners* than previous generations because their world is rich in visual stimuli [Ows97]. Students are spoiled, even dazzled, by the attractive graphical user interfaces of computer software and the possibilities of multimedia applications. Furthermore, teachers are supposed to entertain their students because everything must be fun. This leads to the expression of *edutainment*[2]. As illustrated by Harvard's Graduate School of Education professor Timothy E. Wirth in his keynote speech at the AECT's 2004 conference in Chicago[3]: "Every tool where the student cannot wear a gun is boring". Timothy goes on to say that interfaces of educational tools should be like Alice in Wonderland; have a simple black and white presentation, and the real beauty lies behind the interface.

10.1.3 Teachers and Parents about E-Learning

In traditional learning environments, teachers are primarily responsible for the organization, delivery and assessment of content acquisition by students in their courses. This changes as soon as teachers use E-Learning technologies. Teachers in the cyber-age are often handed the additional roles of instructional designer, technology specialist and administrative advisor.

Bonk [Bon01] interviewed 222 teachers about their position concerning E-Learning technologies. 72% of the respondents publish syllabi and other education material online, and 85% actually use such material in their courses. 71% valued file uploading and downloading tools. 70% use tools for posting cases, questions or problems corresponding to course material on the Web. 57% rated online lecture notes utilities as useful and 44% use online databases in their courses. This confirms Owston [Ows97] who writes that teachers use new technologies like the Internet to increase access to educational resources rather than to improve education. Most of the teachers that do not, or use only little new technologies in their courses say that they do not have the necessary training for mastering that technology [Bon01]. Other issues are: time, technological infrastructure, difficulty in performing lab experiments, and interest.

The availability of online teaching material is increasing, e.g., the tele-TASK archive[4], World Lecture Hall (WLH)[5], the Multimedia Educational Resource for Learning and Online Teaching

[2] `http://www.edu-tainment.org/`
[3] `http://www.aect.org/events/chicago04/`
[4] `http://www.tele-task.de/`
[5] `http://web.austin.utexas.edu/wlh/http://web.austin.utexas.edu/wlh/`

(MERLOT)[6], Deutscher Bildungsserver[7], Kidlink[8], MySchool![9], MIT Open Courseware[10], Explore e-Learning[11], and Learning Science[12]. However, a lot of teachers are still reserved about creating and sharing educational material. The two main reasons are: firstly, course materials are now more mobile than in the past, which raises the sensitive topic of ownership. Secondly, teachers spend more time creating E-Learning content than preparing traditional courses. This supplementary work is often not rewarded. Thus, creating an online course involves more than simply moving an old method of teaching into a new environment [Res05].

In a recent survey [GH07], parents were asked to rank seven ways of learning in order of importance for their child. Only half of all parents selected "classroom lessons" as their first choice, challenging the commonly held assumption that parents always look to school as the center of their child's education. Surprisingly, 4% of parents chose either "surfing the internet" or "playing computer games" as the first or second most important way their child learns. As ever, parents emphasized social experiences with 20% prioritizing either "sharing a meal" or "playing with friends" as their first choice. Two-thirds of parents were certain that their child was "building their general knowledge" through their use of technology. Fathers tended to be slightly more positive about the impact of technology with 47% of men believing their child was developing their creativity compared with 40% of women. Younger parents tended to identify the emergence of less formal skills such as "collaboration" while older parents were more inclined to pinpoint traditional competencies such as "general computer skills".

10.2 Pedagogical Advantages of our E-Librarian Service

Although teaching is much more than only transmitting knowledge, this is the task that can be improved by technological means. In fact, a computer tool cannot explain a difficult topic better than a teacher. It can only present the right information in another form, maybe a clearer, or more exhaustive one.

Our E-Librarian Service fosters autonomous and exploratory learning, it allows the user to ask questions in a very human and simple way, and it returns pertinent and short multimedia answers. In this section, we will describe several pedagogical advantages of our E-Librarian Service. Most of this work was published in [LM04c, LM05a, LM06a, LKM06].

10.2.1 Short Multimedia Clips

Every person is different in her/his sense of perception. Some understand better if they hear the explanation by the means of verbal communication, some need to write it down, others must see it in the form of a text or a picture, and others again have to touch it. A good educational tool must present the same information in different forms in order to activate as many senses as possible. The psychological foundations were proven by the work of Mayer & Gallini [MG90] and Mayer & Sims [MS94]; *information that is presented at the same time in different forms improves the understanding of the information.*

[6] `http://www.merlot.org/`
[7] `http://www.bildungsserver.de/`
[8] `http://www.kidlink.org/`
[9] `http://www.education.lu`
[10] `http://ocw.mit.edu`
[11] `http://www.explorelearning.com`
[12] `http://www.learningscience.org`

The amount of “items” a student can assimilate in a given time depends strongly on his intellectual capacities, and thus is directly related to his age [WMSB01]. Our E-Librarian Service delivers few but semantically pertinent and useful answers to students, so that it can be seen as a kind of *virtual teacher*. In a certain sense this allows students to be taught individually by computers; which teacher can actually do this?

In addition, the length of the clips is essential in our concept. The younger the user, the shorter the time during which (s)he will concentrate on the information displayed on the screen. Therefore, we divided all our multimedia data into small *clips*. The duration of each clip varies from several seconds to three or four minutes. Each clip documents one precise subject or a part of a subject.

Splitting a large topic like computer history or fractions in mathematics into a lot of small pieces was much easier than we assumed at the beginning. We are now convinced that most courses taught in schools or at universities can be divided into smaller atomic units, where each covers one precise subject. Our assertion is based on three topics, for which a knowledge base with small clips was produced; computer history, fractions in mathematics, and computer networks (see chapter 9). Another concrete test was made in biology, where a teacher used our tool to explain the basic function of the heart. Furthermore, teachers of different fields confirmed that this concept could be used in their classes too, e.g., in a language course, a teacher could record one clip per grammatical rule.

One more advantage of that clip approach is the simplicity of administration. If the E-Librarian Service does not cover a certain topic, a new clip can be recorded and added to the knowledge base. The intervention of a computer science expert is not necessary.

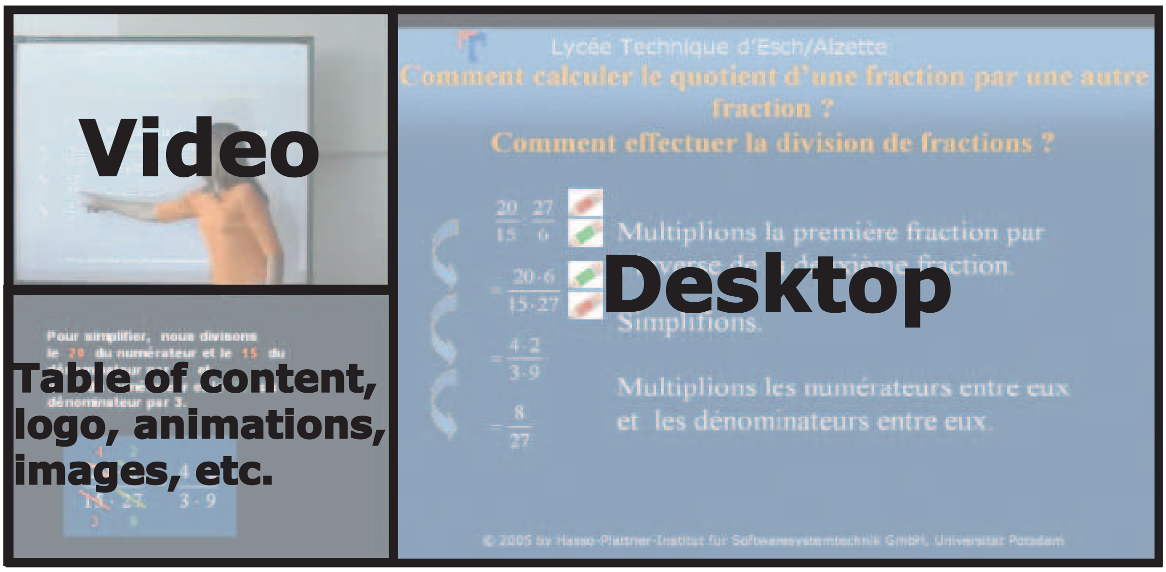


Figure 10.1: Schematic layout of the clips.

All clips in our prototypes were recorded with tele-TASK[13] [SM02]. Basically, the clips are organized in three windows as illustrated in figure 10.1. The first window (video and audio) shows a teacher explaining something on the whiteboard. This is the student’s common view in a classroom, and should create a kind of virtual classroom atmosphere. Based on practical teaching experience we can confirm that students often take lessons, where they use a new computer tool or do research

[13] `http://www.tele-task.de/`

on the Web, as a kind of game, without relation to the normal lessons. The video sequence should keep them concentrated on what they do, and draw their attention to what the teacher is explaining.

The second window represents the content of the presenters desktop, which is simultaneously displayed on the whiteboard that the teacher uses in the video (first window). Although the blackboard is the medium used most frequently in schools, it has many disadvantages such as:

- it is impossible to represent pictures,
- it is difficult and time-consuming for the teacher to create a complex drawing,
- it is time-consuming for students to reproduce its content in their books,
- the content is not available for later lessons, and must be reproduced.

In our clips we use an interactive *SmartBoard*, which offers a lot of advantages compared to an ordinary blackboard, such as:

- the teacher can use this area for an on-screen presentation (e.g., Powerpoint),
- the teacher can add handwritten information to the SmartBoard, which is reproduced in this window both simultaneously and in exactly the same way,
- the teacher can also display the desktop of his/her connected laptop, e.g., to explain a certain application, to show a Web site, or to demonstrate the settings of the computer.

The third window can be used for any purpose, e.g., it can contain links to a photo gallery, hyperlinks to additional information on the Web, book references, or just a single picture of the subject about which the teacher is speaking.

10.2.2 Usability

It is known that students are better able to master, retain, and generalize new knowledge when they are actively involved in constructing that knowledge in a learning-by-doing situation [You98]. Teachers who have used E-Learning tools in their classes report that they changed their teaching style to allow students greater autonomy in their learning [Ows97]. They tend to shift their style of teaching from a didactic, question-answer format to a more exploratory learning approach.

Our E-Librarian Service can be used without special hardware requirements at home or in school, individually or in a group (see section 9). Here are some illustrations:

- At home, our E-Librarian Service can be seen as the student's personal teacher to which (s)he can address questions, and get pertinent and short answers. Today, this kind of "place independent learning" is called *mobile learning*. Hence,:
  - the student can use it to do her/his homework, or to review an important topic before a test,
  - the student can ask for information about topics that were not dealt with in class but which draw the student's attention, or topics of which (s)he needs further explanation for a better understanding.
- We see our E-Librarian Service ideally as a complement to conventional lessons, i.e., in a *blended learning* approach. It is up to the teacher to decide which is an appropriated occasion to use it, e.g.,:

– to introduce a new subject by letting the students discover new information for themselves,
– to use it as a complement to classical syllabuses or the blackboard to find and show illustrations for certain topics in a more attractive form, i.e., multimedia documents.

Information retrieval in multimedia environments actually is a combination of search and browsing in most cases [MSMV04]. Therefore, simple navigation facilities are very important for an educational tool. In our E-Librarian Service, the user can simply browse through the clip, pause the playback at any time, watch it as often as (s)he wants, etc.

In a certain sense, the student creates his/her own course content by assembling different documents. This autonomous and exploratory approach is certainly more motivating for the students, and fosters their sense of responsibility. A similar approach is presented by [QYJ04] where each students gets a special study program according to his/her special demands. In general, more motivated students learn better, and have better results in school.

10.2.3 Human Computer Interface (HCI)

The interaction between computers and humans is still surprisingly complicated. Searching information in a knowledge base means, browsing through an index, or formulating and entering a query normally by entering keywords. In both cases, the user must adapt himself/herself to the machine in order to give precise and machine-readable instructions. However, most people that are not search experts have difficulties, or are not able to formulate their question in a machine optimized way, e.g., by combining search terms with Boolean operators. It is also possible that they do not use the right domain expressions, or make spelling errors. Also, clicking on some icons on the screen is certainly very simple for browsing through the content of a knowledge base, but it is not a very effective way of searching. Finally, forcing a user to formulate his/her query in a computer understandable form is also not a promising solution.

We investigated how to improve this interaction by allowing the user to communicate with the machine in a more human way. We explored the approach of letting the user freely formulate a question in natural language, i.e., by entering a complete question. Instead of typing a question, we could also imagine that the user speaks the question into a microphone.

The graphical user interface must be as simple and ergonomic as possible, especially because we are dealing with an educational tool. An E-Learning interface should neither be too complicated nor too simple: if too complicated, the student gets lost in the menus; if too simple, (s)he could perceive the new tool as a game, and risks not concentrating on the real issue of the lesson. The interface should be adapted to the needs of the user and keep him/her concentrated on what (s)he sees and learns. It is clear that all technical details (e.g., the reasoning and retrieval tasks) must be invisible for the user.

The students' perception and liking of our E-Librarian Service's interface, and how they would accept to enter complete questions instead of keywords, was the aim of a first experiment in an educational environment (see chapter 11).

Chapter 11

Student's Perception of our E-Librarian Service

Contents

Given our premise that the development of our E-Learning Service is not an end in itself but has to be proven useful in a target setting, we asked students to test and to judge our prototype. The experiment and its outcomes will be described in this chapter.

Section 11.1 will formulate the objective and the expectations of the experiment while section 11.2 will describe its organization. The course and the results of the three sessions will be detailed in section 11.3, and discussed in section 11.4. The chapter concludes in section 11.5 with some striking realizations and some learned lessons.

11.1 Objective and Expectations

Given our premise that the development of our E-Learning Service is not an end in itself but has to be proven useful in a target setting, we asked students to test and to judge our prototype. In the following, we will report on these evaluations that we carried out at the Lycée Technique Esch/Alzette (LTE)[1], a technical school in Luxembourg, at the beginning of the year 2005.

Contrary to the experiment described in chapter 12 — which focused on how school results can be improved using our E-Librarian Service — this experiment focused on how the students would accept our E-Librarian Service; a search engine where they have to enter complete questions instead of keywords. In other words, we studied the users' liking of a semantic search engine in comparison to a classical keyword-based search engine; both over the same knowledge base, i.e., computer history, and both having the same graphical user interface (GUI). The key question was: do students realize that the "painful task" of entering a complete question instead of keywords rewards them with a better search result? The experiment and its outcomes were published in [RLM05, LME05].

The used prototype of our E-Librarian Service was CHESt v2 (see section 9.1); at that time the most advanced. It was compared to a keyword-based search engine that we call from now on CHESt v1.

11.2 Organization of the Experiment

In this section, we will start with a description of the test and the participants. Then, we will explain the organization of the three test sessions.

11.2.1 Test Preliminaries

Students from the upper secondary school level (12^{th} and 13^{th} grade) were asked to try out both versions of CHESt — the keyword-based search engine and the semantic search engine — and to provide feedback on the three main characteristics of the tool: the number of results, the pertinence of the results, and the satisfaction with the possibility to enter complete question(s) in natural language (NL) instead of keywords. Three consecutive assessment sessions took place, which differed from each other only in concern of two variables, with one variable being revised from session one to session two, and another variable from session two to session three (see figure 11.1). Additional details about the variables will be provided further below.

11.2.2 General Characteristics of the Three Sessions

For each of the three assessments, a different group of students was to try out both versions of CHESt. None of the subjects had further domain knowledge about computer history. One half of each group started with the keyword-based search, the other half with the semantic search. After 20 minutes, the students were asked their opinion about the tested CHESt version on a number of questions, and then continued within a second trial — again lasting 20 minutes — with the respective other version of the search engine. In order to provide the subjects with some general context within which they could search for information, three questions (in the following named "frame questions") were presented at the beginning of each trial (i.e., six questions per task for each student; see further below for some examples of frame questions).

[1] `http://www.lte.lu/`

	Type of frame questions	Instructions given about how to use the search engines
Session 1	Precise questions, e.g., “Who invented the Z3-machine?”	Students were instructed to enter single or multiple words, the way they thought they would obtain the most pertinent results.
Session 2	General questions, e.g., “Describe the early years of the Internet.”	Idem session 1
Session 3	Idem session 2	Students were told to enter questions while using the semantic search, and keywords while using the keyword-based search engine.

Figure 11.1: Settings of both variables for the three assessment sessions.

At the beginning of each session, the students were informed that two search engines allowing to search for information from the domain of computer history would be presented to them. They were told that the aim of the session would not be their successful answering of the frame questions, but rather their personal judging of the efficiency and their general liking of the respective search engine. They were also briefed that the GUI would be the same for both versions, and that no questions would have the GUI as target. The students were informed that their main job would consist in judging whether the search results yielded by the respective search engine would match their queries, and whether they really found the information they had been looking for. After the respective version of CHESt had been tested, the students answered questions focusing on the following issues:

- Did the tested search engine yield too few, too many, or an adequate number of results? This question aimed at clarifying the personal judgment concerning the quantity of the results. Students might actually find what they searched for, but they might have expected more results.
- Did the search results exactly fit the queries? This question aimed at knowing whether, in general, the subjects had the impression that the result(s) listed was/were pertinent in regard to the keywords or questions they entered within the search field.
- Did the subjects find the information they have been searching for? This question is considered separately from others about the general fitting of the results, as the user might have found results that fitted the queries well, but still might have been unable to find what (s)he had actually been looking for.

After both versions of CHESt had been tested by the subjects, they were questioned on the following issues:

- Some of the questions asked for a comparison between both versions, aiming at finding out whether the participants had the impression that the one or other version would provide the more fitting results.
- The users were also asked which version they would choose if they had an exam within the domain of computer history during which they were allowed to use one of the CHESt versions.

This was to find out about the general preference for one of the two versions within a concrete context.

- Finally, one question raised the issue of the students' opinion about having the possibility of asking complete questions instead of keywords.

11.3 The Course and Results of the Sessions

In this section, we will describe the course and the results of the three consecutive sessions. They differed from each other only in concern of two variables, with one variable being revised from session one to session two, and another variable from session two to session three (see figure 11.1).

11.3.1 First Session

18 male students from the 13^{th} (terminal) grade of secondary school (technical formation; mean age 21.25) participated within this first evaluative assessment. No information was provided about the difference between the two search engines; students were instructed to enter single or multiple words, even complete questions, just the way they thought they would obtain the most pertinent results. Furthermore, the participants received precise frame questions such as the following ones:

- Did Howard Aiken and Charles Babbage know each other?
- Find three interesting inventions from the software domain (e.g., operating system, programming language, or application). Which company or person has invented this and when was it published?

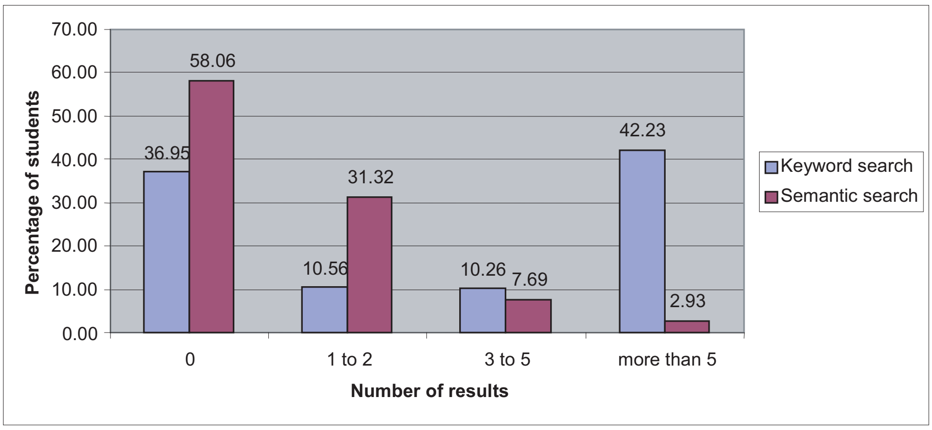


Figure 11.2: Number of results per CHESt version: percentage of total number of results.

Asked about the number of the yielded results, the majority of the students think there is either an adequate number of results (7 students) or even too many results (7 students) generated by the keyword-based search. Meanwhile, considerable 14 out of 18 students asserted that the semantic search engine yielded too few results. The real number of results generated by the respective search engines (see figure 11.2) confirms that a higher percentage of queries within the semantic search than within the keyword-based search yielded no results. In the meantime, the keyword-based search led to more than five results in 42% of the search initials.

The question about the pertinence of the yielded results revealed an obvious superiority of the semantic search function. While 78% of the subjects said that in only a few or in approximately half of the cases the keyword-based search would have provided fitting results, 78% considered that "most of the results" (61.1%) or "all of the results" (17%) fitted the search subjects within the semantic search engine.

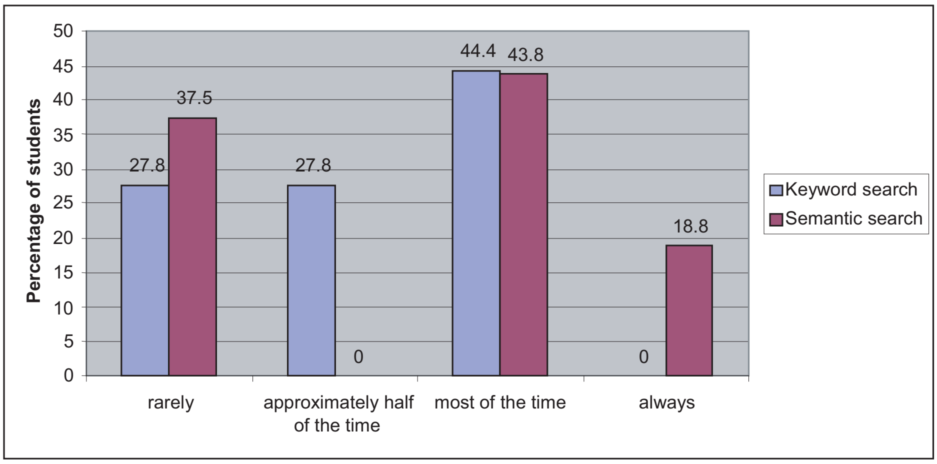


Figure 11.3: Within the listed results I found what I have been looking for.

There was not a similar obvious difference between the two search engines concerning the question whether students found what they were looking for. The subjects judged the efficiency of both search engines quite similarly (see figure 11.3). Why is there this small discrepancy between the fitting of the results and the success in finding what was looked for? If taking a look at those last two questions in combination, one can observe that the incongruity is especially due to only a few of the students. Actually, for the keyword-based search, only one student meant that, although all of the results fitted the keywords, he only found what he looked for in approximately half of the search initials. Concerning the semantic search, three students (out of 16) said that most or all of the results fitted their search subjects, but that they still only rarely found what they had been looking for.

		Frequency	Percent	Cumulative Percent
Valid	1	5	27.8	27.8
	1 and 4	1	5.6	5.6
	2	7	38.9	38.9
	2 and 4	1	5.6	77.8
	3	2	11.1	88.9
	4	2	11.1	100.0
Total		18	100.0	

Figure 11.4: What version did the users prefer? Choice of the version 1=keyword-based search, 2=semantic search, 3=both versions equivalent, 4=none of the versions.

Asking students which one of the search engines they would prefer if they had an exam on the subject of computer history, the answers given were not clearly pointing in one or another direction (see figure 11.4). Although most of the students were fairly satisfied with the fitting of the results from the semantic search engine, and although there was no greater difference between the keyword-based search and the semantic search in terms of finding what has been searched for, only 39% would choose the semantic search engine, compared to 28% announcing their preference for the keyword-based search, and 22% not deciding on any of both versions.

Finally, when asked about their liking of the possibility of entering whole questions instead of single keywords, half of the students (N=9; 50%) indicated that this possibility is only considered to be advantageous if it also yields better results than a keyword-based search.

To summarize, the first evaluation session revealed that, although most of the subjects were rather satisfied with the fitting of the results provided by the semantic search engine, they were not completely convinced of the (possible) advantages of the semantic version of CHESt.

Before discussing these results in greater detail, the realization and results of the two other evaluative sessions shall be described. The principal aim of the subsequent session was to replicate the results of the first session with more general frame questions. Indeed, the analysis of the keywords and sentences entered showed that most of the students were sticking all too strictly to the respective frame questions in their formulation of the questions and keywords in the question bar. In order to investigate whether similar results would be obtained when the students are given greater liberty in their searching for information on the subject of computer history, more general tasks were formulated for the second and the third evaluation session, described in the following sections.

11.3.2 Second Session

18 students (17 male) from the 12^{th} grade of secondary school (technical formation; mean age 19.76 years) participated within this second evaluative session. This time, the frame questions were more general than in the previous session; examples of frame questions are as follows:

- Give an overview of the last 60 years of computer evolution.
- Explain why, especially around World War II, computers had been developed. Name three examples of such computers and their respective inventor(s).

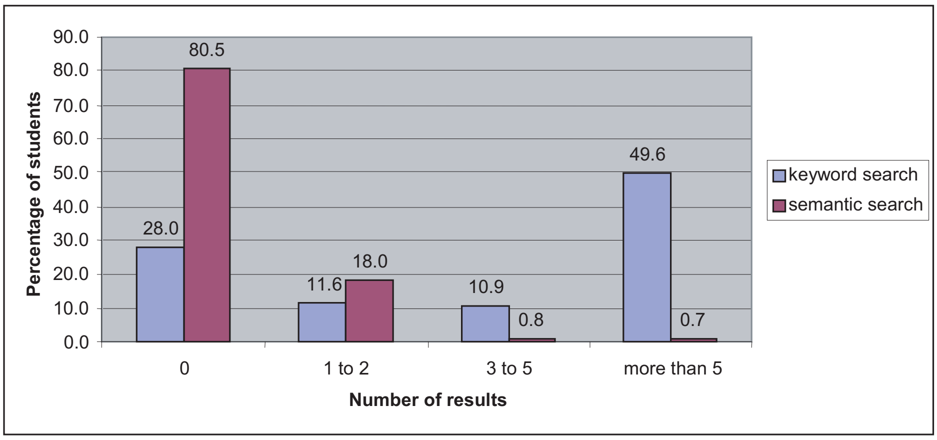


Figure 11.5: Number of results per CHESt version: percentage of total number of results.

In correspondence to the results from the first assessment, the students in this second session showed to be more satisfied with the number of the results listed by the keyword-based search engine. 56% asserted that an adequate number of results were listed by the keyword-based search engine, only 17% said it had been too few. This can be contrasted with respectable 78% of the students thinking the semantic search had generated too few results. The mean number of results yielded by the semantic search engine is way below the one in the first session, with no results yielded at all in considerable 80.5% of the search initials (see figure 11.5). The keyword-based search in the meantime yielded no results in only 28% of the searches, while half of the search initials (50%) led to more than five results.

This time, the question about the pertinence of the yielded results revealed a different answer pattern than within the first testing session. Actually, the keyword-based search was judged more positively than in the previous testing, while the semantic search engine was given a worse evaluation. This led to the results pattern of a nearly equivalent number of students (9 and 10, for the keyword-based and semantic search, respectively) asserting that "most of the results" or "all of the results" fitted the search keywords. Two students asserted not having had any fitting results at all during the semantic search (while none of the users did so concerning the keyword-based search).

As far as the question of whether the students were able to find what they were looking for is concerned, a rather negative picture is being presented for the semantic search. Three people never found what they were looking for, and four users were only rarely able to locate some interesting information. This is the case "only" two times within the keyword-based search. Clearly most of the students (76%) confirmed being successful in finding the information they had been looking for with the keyword-based search, compared to 47% concerning the semantic search engine.

If comparing and combining the answers to these last two questions (fitting of the results and finding what one has been looking for), we noticed that only for very few of the students there was an incongruity between the judgment on the general fitting of the search results to the entered queries on the one hand, and the judgment about whether the searched information was found on the other hand. For the keyword-based search one student (out of 17) believed that, although most of the results fitted the keywords, (s)he rarely found what (s)he had been looking for. Two students asserted that, although only few of the keywords fitted the results, they still found what they had looked for in most of the search initials. Concerning the semantic search, a similar pattern of the results is obtained, with two students saying that most or all of the results fitted their search words, although they only rarely found what they had looked for. One user indicated that although only few of the keywords fitted the results, (s)he still found what (s)he had looked for in most of the search initials.

		Frequency	Percent	Cumulative Percent
Valid	1	17	94.4	94.4
	2 and 4	1	5.6	100.0
Total		18	100.0	

Figure 11.6: What version did the users prefer? Choice of the version 1=keyword-based search, 2=semantic search, 3=both versions equivalent, 4=none of the versions (more answers permitted).

Even though the previous questions already revealed that users did not have such an obvious judgment in one or the other direction about one of the CHESt versions any more, asking them which version they would prefer to use during an exam in the domain of computer history revealed an obvious preference of nearly all of the students (17 out of 18) for the keyword-based search (see figure 11.6).

Approximately half of the students (10 out of 18) finally stated that the possibility to enter whole questions instead of single keywords would be a good option only if this would lead to better results than with keywords.

In summary, the second evaluative session — using more general frame questions and thus providing the participants a greater liberty in their information search — revealed a slightly more negative judgment of the semantic search engine than the first session. Firstly, the students indicated being more satisfied with the number of results provided by the keyword-based search than those by the semantic search. Secondly, although the judgment about the fitting of the listed results was comparable for both CHESt versions, more users pointed out finding what they were searching for within the keyword-based search rather than with the semantic search. The clear preference for the keyword-based search finally underlines that the characteristics of the keyword-based version of CHESt are the more appealing ones.

One final point still remains to be pointed out in this context: actually, the analysis of the log-file of this session reveals that students did not refer very often to the possibility of combining keywords and/or even entering whole questions, which would ensure optimal "communication" with the semantic version of CHESt. We hypothesized that the high number of semantic search initials yielding no results at all is due to this problem. In order to investigate whether the rather negative results concerning the semantic version of CHESt are the result of whether users do or do not fully exploit the potential of CHESt, a third evaluative session took place in which the explicit instruction was given to enter complete questions in the search field when using the semantic version of CHESt.

11.3.3 Third Session

14 students (11 male) from 12^{th} grade of secondary school (general technical education; 17 to 20 years old) participated in this third assessment. While the frame questions remained rather general (as within the second session), the students were, this time, explicitly told to enter complete questions while using the semantic search, and to enter single or multiple words while using the keyword-based search. The judgments given by the students during this third session were still comparable to the ones provided in the second assessment, as outlined in the following section.

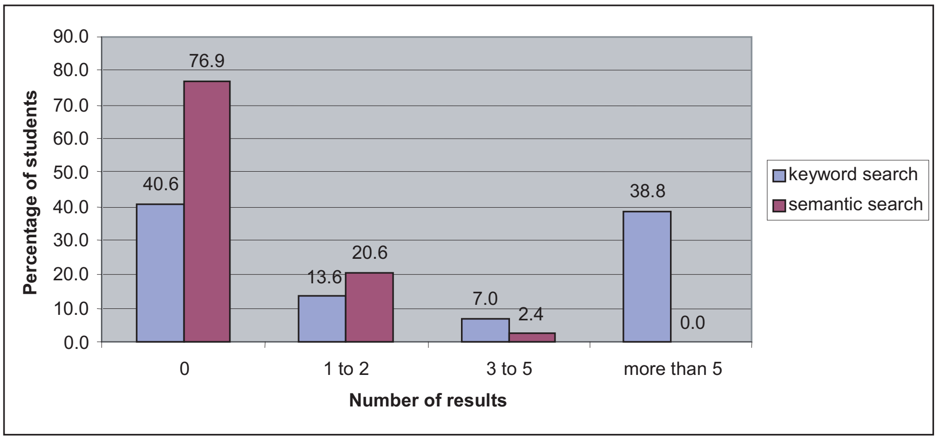


Figure 11.7: Number of results per CHESt version: percentage of total number of results.

Comparable to the results from the previous assessment, the students were more satisfied with the number of results listed by the keyword-based search than with the number generated by the semantic search. While 64% asserted that an adequate number of results were listed by the keyword-based search engine, only 21% said so with relation to the semantic search engine. 79% had the opinion that the semantic search listed too few. The analysis of the mean number of results yielded by the respective queries further consolidates the finding out of a very elevated percentage of semantic search initials not having generated any results at all (77%) during the previous sessions (figure 11.7).

The overall satisfaction with the fitting of the results listed by the semantic search engine was even lower than in the previous assessments. While 57.1% affirmed that most of the results fitted the words they entered when using the keyword-based search, only 33% (25% saying "most of them", 8% saying "all") said so as to the results yielded by the semantic search.

The same is true for the success the users experienced concerning the finding of what they were looking for: with the keyword-based search, eight people (out of 13) asserted having found what they were looking for "most of the time" or "always", but only two people were equally confident concerning the semantic search engine. No greater discrepancies were found as to these two questions, with only one student asserting that, although only few of the results fitted the queries, (s)he still found what (s)he had looked for most of the time (both CHESt versions).

Asking students which version they would prefer to use during an exam in the domain of computer history revealed an obvious voting of the great majority of the students (11 out of 14) for the keyword-based search. Finally, 79% of the users emphasized that they would like the option of being able to ask complete questions instead of keywords only if this yielded the better results.

In summary, this last assessment session left us with a rather devastating image of the semantic search engine. Firstly, students were again not as satisfied with the number of results provided by the semantic search as those listed by the keyword-based search. Also, even when giving concrete instructions as to how to use the search engines in order to guarantee that all qualities of the semantic engine are exploited, a very high percentage of the queries still remain without any yield of results. In addition, even fewer students than in the previous sessions really seemed to be satisfied by the pertinence of the listed results. The same is true for the judgment of the success in finding what the users were looking for. This third assessment session thus confirms and consolidates the finding that several of the qualities of CHESt have to be revised in greater detail.

11.4 Discussion and Conclusion

The present investigation aimed at evaluating the qualities of the keyword-based and the semantic version of the E-Learning tool CHESt in an educational environment. Students from the upper secondary school level were asked to test both versions and judge them on the number and the pertinence of the yielded results, as well as to give their opinion about the possibility to enter questions instead of keywords. The results from the three evaluative sessions especially revealed two things in particular: students prefer the keyword-based search engine, and our E-Librarian Service yielded often no result at all.

11.4.1 The Users' Liking of the Search Engine

The first striking result was that the subjects generally preferred the keyword-based search to the semantic search. This was found to be independent from the judgments about the appropriateness of the yielded results. Even in the first session, where the majority of the users claimed that the semantic version yielded pertinent results, compared to only a few concerning the keyword-based search, most of the users decided to use the keyword-based version of CHESt within an exam on computer history.

There are multiple reasons that could provide an explanation for this result. Actually, the students could just be more accustomed to use keyword-based search engines (nearly all search engines on the Web are keyword-based). Further, entering keywords might have been experienced as an easier and more comfortable task than entering whole sentences. Blondel [Blo01] concludes in his survey that the students' preferred method to query search engines is without any doubt by keywords.

In the light of the generally valid claim of the users that they like the option of being able to ask complete questions instead of keywords, but only if this yields better results, we still realized that the supplementary intellectual task of thinking and formulating whole questions does not necessarily have to be considered as a real burden compared to the entering of single, maybe only loosely related keywords.

Finally, the number of results generated by the respective search engine is considered to be a factor of central importance concerning the rating of the CHESt-versions. Actually it turned out that, more than half of the semantic search initials did not lead to any result at all. In contrast to this, the keyword-based search generally listed a high number of results (judged as being "too many", or — more often — as being "neither too many, nor too few"). Seeing the differences within the number of results that were in the mean yielded by a keyword-based and a semantic search respectively, the users might have experienced many more opportunities to explore the content of the knowledge base with the keyword-based rather than with the semantic search. The number of the yielded results is seen as a characteristic that might be of central importance, especially if users of an E-Learning tool do not yet have an elaborate knowledge within the domain of interest (such as computer history). The fact that during the last two sessions users more often found what they had searched for with the keyword-based than with the semantic search engine, further underlines the advantageous possibilities the students might have experienced with the higher number of results produced by the keyword-based search.

11.4.2 Pertinence of the Search Results

One second striking finding concerns the pertinence of the results. During the first evaluative session, the subjects were more confident in this concern while using the semantic search engine. As opposed to this, subjects judged the keyword-based search more positively in this regard during the other two sessions, where more general frame questions had been given. Thus, providing the users with additional freedom to explore the knowledge base has led to search queries that provided less convenient results than queries that were initiated within a more restrictive context. We refer to three explanations concerning this pattern of results.

First of all, this finding might be explained by the fact that users directly associated this question with the one about the number of the results: as a very high percentage of the search initials didn't yield any results at all during the semantic information queries, the judgment on the pertinence of the results might have been strongly influenced by this (with an interpretation such as "no results at all means no fitting").

Secondly, the difference in this matter between the first and the other two sessions might suggest that students have had general difficulties to formulate questions of their own in order to explore a domain such as computer history. This might have resulted in a general sticking to the rather specific frame questions during the first session. Indeed, it was possible to take exact questions/words that were given as frame questions (or to slightly change the formulation of those) and to enter these within the search field. Such a strategy was not possible any more to the same degree during the two subsequent sessions, where users were expected to think about and formulate questions more autonomously.

Thirdly, this finding of a difference between the first and the other two sessions however might also suggest that we have to improve the semantic search engine concerning its main characteristic: the understanding of the users' questions. As already outlined above, although many of the yielded results were judged to fit the search subjects, just too many queries have yielded no results at all. Given the fact that both search engines access the same knowledge base, and that the judgment about the pertinence of the results did not differ significantly between sessions two and three (where emphasis was placed on the instruction to enter whole questions) this finding underlines that future efforts will need to focus on the improvement of the semantic search engine.

11.4.3 Conclusion

In conclusion, the results of our assessments suggest that the satisfaction users experience with a search engine like CHESt (in its two versions) seems to strongly depend on three factors.

- The practice users have with the respective search engine (with the formulation of questions). Users need guidance in how to formulate effective queries, especially if they are not expert in the focused domain [FDD+99, Blo01, NPSR99, CMZ03, Mor05].
- The background knowledge users have concerning the focused domain (here: computer history); little knowledge within a domain of interest seems to require good basic opportunities (such as a list of domains to be explored or a search tree) to explore a given knowledge base.
- The factual success of a search engine to find the requested content — which, again, might depend on the content of the knowledge base.

11.5 Striking Realizations and Lessons learned

Firstly, let us point out that nearly all students approved the appealing multimedia presentations. They agreed that the explanations were sufficiently complete to understand the subject. Several appreciated the short length of the clips; few stated that the clips were too long. Some proposed that the system should also offer a textual version of the clips, which could be copied into a word processor. Some added that they appreciated the short response delay of the system.

None of the students seemed to plan his/her search. In fact, the majority of all users started to enter the questions in exactly the same way as they were presented, or they entered keywords that were used in the formulated questions. The semantic search was able to return satisfactory results for the students of group A, because they had more precise questions. However, for general questions (group B and C), the semantic search was not able to find an appropriate clip and returned no result. The interactive nature of CHESt supported the student's belief that there was no need to plan ahead because the progression of a search would be largely determined by what they saw on the screen. This theory is confirmed by [FDD+99, Blo01, Mor05]. Similar explanations were given in [NPSR99, HS00]; they found a clear difference in strategy depending upon the experience of the participants. Novices, as in this case, typically start with very general queries — where CHESt was not able to find an appropriate answer — and gradually narrowed down the search, adding and changing words in the query. Lau & Horvitz [LH99] analyzed and modeled the log of the Excite search engine and found out that most actions were new queries, and relatively few users refined their searches by specialization, generalization, or reformulation.

One of our most interesting realizations is that the very large majority of students turned out to use only keywords to formulate queries, independently of the used search engine; even the students in group C, who were told to enter complete questions in the semantic search engine, did so. The log files showed that several students of group A and B, and all the students of group C, tried one or more complete questions, but then switched to keywords. When asked later why they only used keywords, they explained that by entering questions, the semantic search often did not return a result, and that they had been more successful with keywords. Asked if they were aware that their question might not be precise enough, or if they did not think about reformulating their question, they replied no. Several students commented that they were simply more at ease with a keyword-based search and that entering keywords was easier and required less effort. In fact, all students noticed that the keyword-based search generally returned a result, whereas the semantic search frequently did not understand the user's question, and thus returned no result. We observed that students who entered questions became frustrated quickly if no immediate result was returned, and then changed their search technique by entering keywords for the rest of the experiment.

An interesting observation was that when a search returned no result, students tended to re-enter a previous question (or keywords), where they were sure to get a result. [FDD+99, Blo01, HS00] found out that students often return to "landmarks" where they received good answers. Also, the re-appearance of information in the clips may have helped some students to remember certain facts [HPM05].

We observed that a lot of students judged the pertinence of the system's answer and the quality of the search engine by the number of results that were returned. On the one hand, if the semantic search engine returned just one (maybe the most pertinent result), some students complained that there were not enough results. However, most appreciated a short list of results. On the other hand, if a search returned too many results, most students quickly browsed through the list and tried another query. Some students launched different queries without consulting any answer, only to reduce the number of results. Similar results were found by [Blo01, HS00].

Asked how they chose a clip among the list of results (especially in case of a keyword-based search), the majority of the students were not able to give a precise explanation. Most of them selected one clip randomly, started to watch it and decided relatively quickly if that answer was pertinent. This confirms the results of Fidel et.al. [FDD+99] that students make quick decisions about whether or not a document is relevant. Students have clear expectations about the requested search result. A result was often not accepted if that clip only contained some pieces of the expected answer. Students wanted one document to include all the information they needed.

Finally, 22% of the users would prefer to enter complete questions instead of keywords, 69% would prefer to enter complete questions instead of keywords only if this yielded better results and 8% would dislike this option.

Chapter 12

Better Results in Mathematics with MatES

Contents

In pedagogy, it is very difficult to prove the gain of an educational tool; it is not possible to have the same students with an "initialized" memory for different tests. However, in this chapter, we will show that E-Learning can improve school results. Our assertion will be based on the outcomes of an experiment that we made in a school with MatES, a prototype of our E-Librarian Service.

The objective and expectations of the experiment will be formulated in section 12.1. Section 12.2 will provide details about the organization and the course of the experiment. The outcomes will be detailed in section 12.3, and discussed in section 12.4.

12.1 Objective and Expectations

The objective of this experiment is to test the advantages of our E-Librarian Service in a normal educational environment, and to investigate in how far this alters students' school results positively. The prototype MatES (see section 9.1.2) was specially developed for this experiment. The outcomes were published in [LDM06, LDM07].

There was no classical lesson — i.e., teacher-centered lesson — during five weeks, where the teacher gave explanations, but the students had to learn in an autonomous and exploratory way. They had to ask questions to MatES just the way they would if there was a human teacher.

Our hypothesis was that this different training approach (where each student is active in the learning process and plays the role of an explorer) would result in higher motivation, and produce students, who are willing to put more effort into learning mathematics. Furthermore, it was to be investigated if the simple multimedia presentations would be complete enough for the students to acquire enough knowledge to understand a certain subject without the help of the teacher.

12.2 Description of the Experiment

In this section, we will give an overview of the experiment. First, we will describe how we divided the class in three groups. Then, the course of the different lessons — lessons before the experiment, first lesson of the experiment, the course of the other lessons, and the course of the tests — will be described in detail.

12.2.1 Grouping the Students in Three Clusters

22 students, aged between 12 and 14 years (7^{th} grade) from the Lycée Technique Esch/Alzette (LTE)[1] took part in the experiment, which lasted 5 weeks; from February, 13^{th} 2006 until March, 16^{th} 2006. This is the normal amount of time spent on teaching fractions in this grade. All lessons took place in a computer room.

In the first term of the school year (September, 18^{th} 2005 until February, 12^{th} 2006), the students learned geometry (volumes, surfaces, etc.). Four class papers were written about that subject.

All students already had some basic knowledge about geometry and fractions because these subjects had already been introduced in the three previous school years. We made a preliminary test at the beginning of the experiment (February, 9^{th} 2006) to measure their current knowledge about fractions. The students were not prepared for this test.

Based on the results of this preliminary test, and the results on geometry (first term), we grouped the students in three clusters (see figure 12.1): weak (8 students), middle (6 students) and strong (8 students). This classification helps us to evaluate our style of teaching according to three initially different levels of knowledge. We suppose that generally weak students will also have problems to learn fractions, and that strong students will also do well in fractions. We will investigate in how far using MatES will alter the configuration of the clusters.

The graphical representation that we use is based on the theory of hypervolumes [Har96]. For a given number of clusters (here: 3), the aim is to link the points to form convex figures (as many as there are clusters) so that the sum of the surfaces of all figures is minimized.

Figure 12.1 shows that there is no relation between the preliminary test and the results on geometry. Some strong students did well in the preliminary test, others not at all. Some weak

[1] http://www.lte.lu/

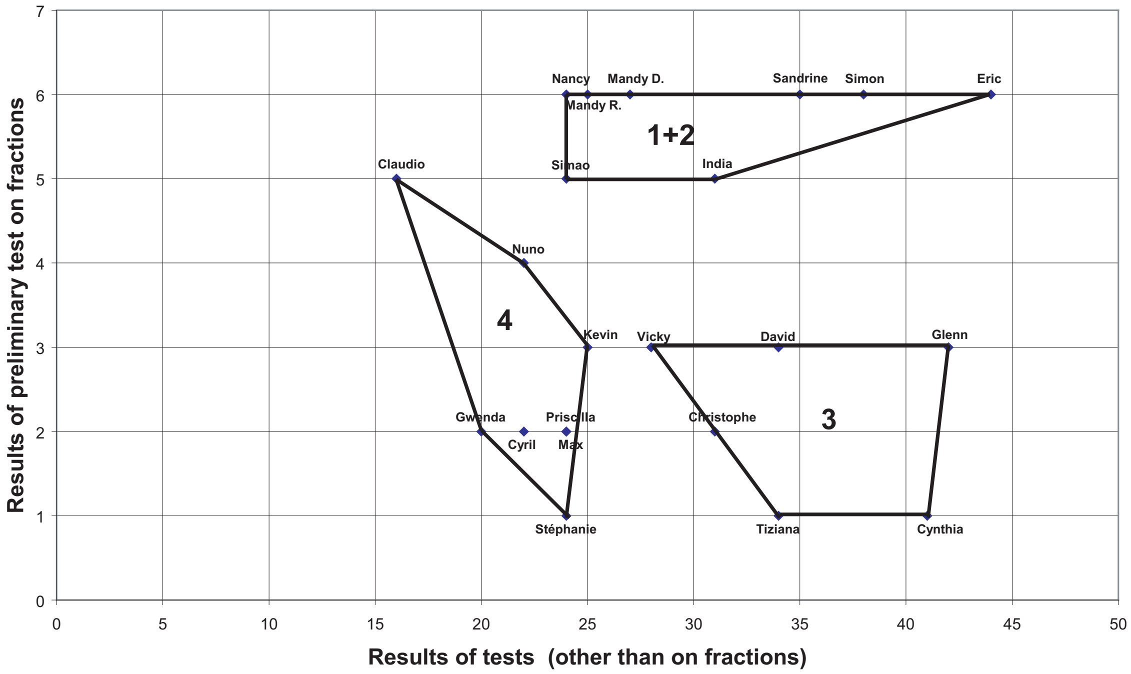


Figure 12.1: Average of the four tests about geometry (x-axis) and the preliminary test about fractions (y-axis) for the three clusters of students.

students did well in the preliminary test, others not at all. This shows that their current knowledge about fractions was completely heterogeneous.

12.2.2 The Lessons Before the Experiment

In mathematics the teacher classically introduces a new topic on the blackboard. Then, exercises are trained together; normally one student or the teacher is on the blackboard and writes the solution. However, in our experiment there was no "theory" about fractions that was explained. We broke the "didactic contract", and employed a different pedagogy; we let the students play the role of an explorer who had to discover and to acquire new knowledge all by themselves in an autonomous way, and by using MatES as a kind of personal teacher.

In the lessons before the experiment the teacher explained her intentions to the class, and gave an example how they would work together. She explained that they would get exercises each lesson, and that they would have to find appropriate clips to watch in order to acquire the knowledge they needed to do the exercises. The students had to make the choice, which clip to watch.

The aim was to make the students understand that this kind of teaching would be more adapted to their cognitive capacities, allowing slower students to watch the same information as often as they want. Furthermore, this kind of teaching enables the teacher to guide and assist them in a more personalized way.

12.2.3 The First Lessons

The issue of the first lessons was to figure out that it is difficult to get good information quickly, and that this is especially true with a keyword-based search engine.

In the first lesson, the students learned how to formulate correct questions, because this is not easy to do for students of the 7^{th} grade. Furthermore, the teacher explained the advantages of entering complete questions in a *semantic search engine* compared to a simple keyword-based search. The number of results was indeed a compelling argument to use MatES; e.g., the same question, "How to Simplify a fraction?", in Google yielded 31400 results, whereas MatES yielded only two.

In the second lesson, the students learned how to use MatES. In practical exercises they used MatES to train vocabulary on fractions. The teacher gave a sentence with a missing word, e.g., "We need to learn fractions because they represent...". The students' task was to formulate a question, and to find the missing word by watching the yielded clip(s), e.g., "Why do we need to learn fractions?", or "What do fractions represent?".

12.2.4 The Course of the Other Lessons

The course of all lessons during the experiment was the same: the students did exercises, and had to acquire the missing theory autonomously.

At the beginning of each lesson, they got a sheet with exercises to solve. Their first task was to find out what they already knew and what they did not know to solve the exercises. Then, they had to ask questions to MatES, and had to watch the yielded answers (clips) to complete their knowledge. The teacher was always present and helped if a student did not understand an explanation, or still had problems solving the exercise. Only some examples of exercises were briefly developed in class to illustrate some general mistakes or misunderstandings.

The level of difficulty of the exercises was different for the three clusters. Strong students got more advanced exercises, and weak students got very simple exercises. This allowed all the students to progress at their own pace, to be permanently occupied, and not to feel overwhelmed by the degree of difficulty of the task at hand.

The teacher reviewed all exercises at home. She marked mistakes and suggested important subjects to consider. This allowed the teacher to continuously evaluate the class and to keep a hand on the experiment (e.g., to stop it, if something turned wrong).

12.2.5 Course of the Tests

Two tests were written on fractions; the first in week 3, and the second in week 5 of the experiment. Both had the same type of questionnaire, and were of the same level of difficulty. The first test was more about basic subjects (e.g., the representation of fractions), whereas the second one was about operations on fractions (e.g., the sum of two fractions). Each test lasted two hours; one hour for a *classical test* (30 marks) and one hour for a *practical test* (30 marks). As for the first one, the test took place in a normal class room under classical conditions: no books, notes, calculator, etc. were allowed. The students received a questionnaire, and had to write the solutions onto a blank sheet. The exercises were based on the knowledge they had (to have) acquired autonomously during the past lessons.

After one hour, the students moved to a computer room and continued with the second part of the test, the practical one. Each student worked individually on a computer. They received a questionnaire, and had to write the solutions onto a blank sheet. Contrary to the first part of the test, these exercises were about unknown concepts (e.g., proper fractions). Therefore, the use of MatES was implicit.

Nearly all students finished both tests in time. There were also no significant complaints about the tests, regarding an exaggerated level of difficulty.

12.3 General Results

In this section, we will describe and discuss the outcomes of the experiment. We will start with the students' results in the tests, report on interviews with the students about their impressions, and analyze the log-files. We will conclude with some general observations.

12.3.1 Students' Results

No significant differences could be measured between girls and boys, for either the geometry or the fraction results.

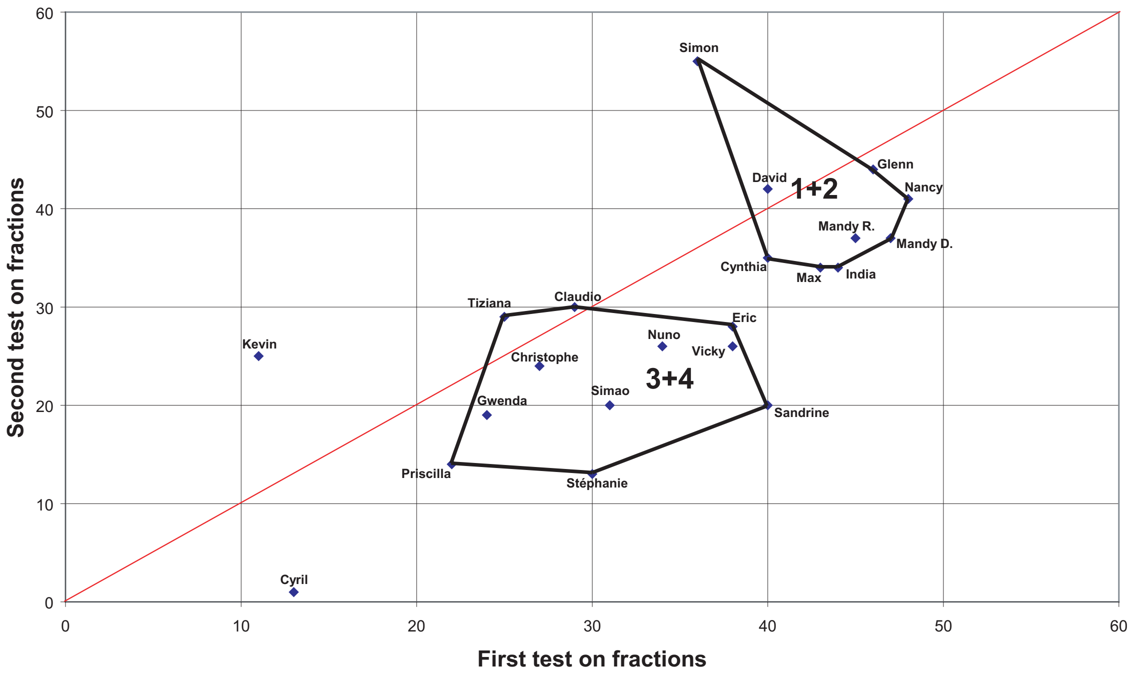


Figure 12.2: Results of both tests about fractions: 1^{st} test (x-axis) and 2^{nd} test (y-axis).

Except for two students, the class could be grouped in two clusters (see figure 12.2): the students who did well in both tests (1 + 2), and those who had worse results in the second test (3 + 4). In general, all students did (very) well in the first test. However, the results of the second test were not so positive. Though, the second test was not more difficult than the first one, it took place in a time when the students had a test almost every day. This could explain the worse results in the second test.

There were also some interesting differences between the two parts of the test: the theoretical- and the practical one. The differences were not significant in the first test, but they were much stronger in the second one; the results in the practical part were better than those in the theoretical one. An explanation could be that as for the first part (the theoretical one), the students were tested on their theoretical knowledge about fractions and how well (or badly?) prepared they were for the test. But as for the second part, the exercises were about unknown concepts so that even the students who did not learn a lot for the test, could get a good result by asking questions to MatES.

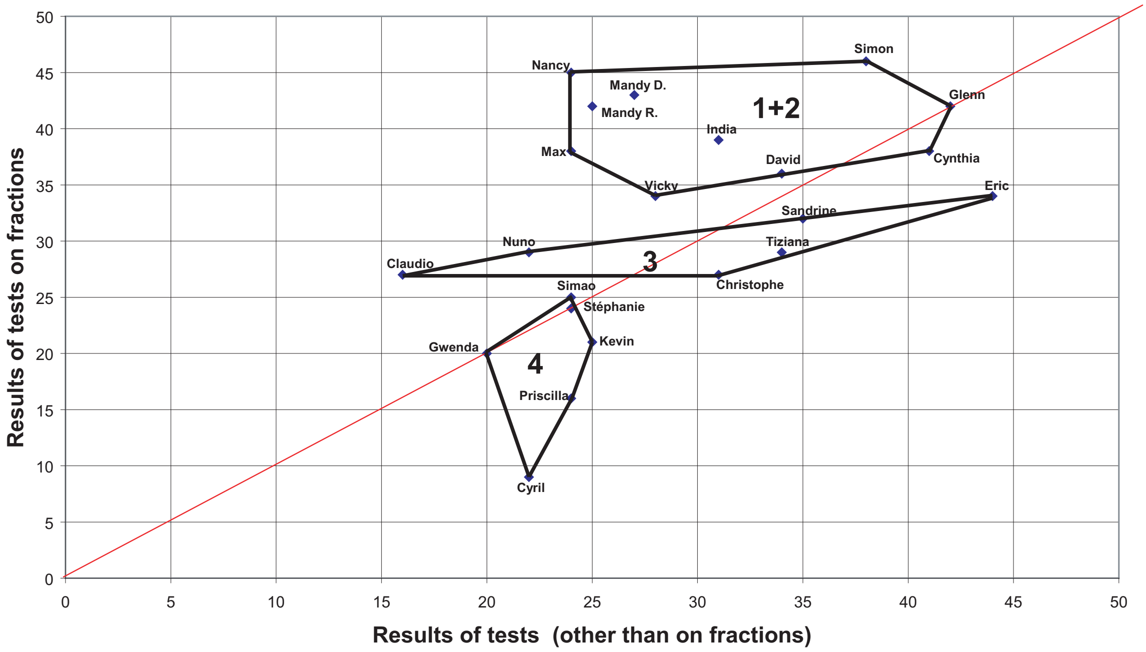


Figure 12.3: Average of the tests about geometry (x-axis) and the tests about fractions (y-axis).

A lot more revealing is the comparison of the results about fractions with the results about geometry (see figure 12.3). We can make three general assertions: about the general results of the tests, about the changes in the clusters, and about the proximity of the clusters.

Firstly, the overall results were better on fractions (average result of the class: 32/60) than on geometry (average result of the class: 29/60), which represents an improvement of 5%. This number was confirmed by a t test of means with two paired independent samples. 11 students had better results in fractions than in geometry (they are located in the graph above the identical function). 9 of them progressed very much (at least 6 marks with a maximum of 60 marks for a test). There is even one student whose progression is 21 marks. 8 students regressed, 3 of them very much (at least 6 marks). 3 students stayed constant. In total the 11 students progressed by 111 marks against the 8 students that regressed by 50 marks.

Secondly, the composition of the clusters changed positively. Before starting the experiment, the clusters were composed like this:

Cluster	Number of students	Percentage
weak	8	36.4%
middle	6	27.3%
strong	8	36.4%

Here is the composition of the clusters at the end of the experiment:

Cluster	Number of students	Percentage
weak	6	27.3%
middle	6	27.3%
strong	10	45.5%

7 students progressed in a higher cluster with one student who progressed by 2 clusters (from cluster “weak” to cluster “strong”). 2 students regressed from cluster ”strong” to cluster “middle”, and 1 student regressed by 2 clusters (from cluster “strong” to cluster “weak”). The latter has nevertheless better results on fractions than on geometry. 12 students stayed in the same cluster (5 in the cluster “weak”, 2 in the cluster “middle”, and 5 in the cluster “strong”). Generally, the weakest students stayed weak, and strong students stayed strong. Therefore, the major changes were in the cluster “middle”. Those 3 students (out of 6) that remained in the cluster “weak” generally had bad marks in all branches; they did not even maintain a correct exercise book. It seemed that they resigned completely.

Thirdly, by comparing figure 12.1 and figure 12.3 one can observe that before the experiment the knowledge of the class was generally very heterogeneous. After the experiences with MatES, their knowledge became more homogeneous; the difference between strong and weak students was less significant. Indeed, the dispersion graphs (see figure 12.4) show that there exists a weak linear relation, and a polynomial adjustment. The points of these graphs are less dispersed than those of figure 12.1. We think that this is mainly due to the fact that the students worked autonomously and saw the sense in the activities they did.

12.3.2 Students’ Impressions

This evaluation is based on one written survey (end of week 1), weekly general discussions, and mostly one individual interview session (end of week 5).

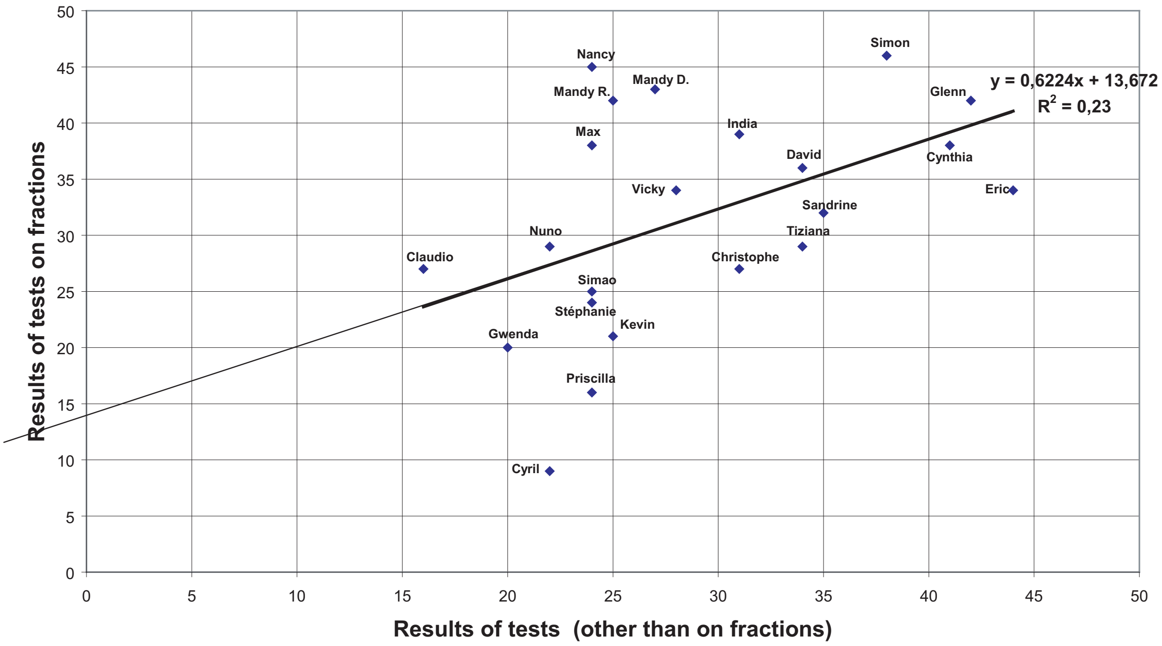
12.3.2.1 Comments About this Kind of Learning

When asked if they think that they learned better with MatES compared to classical teaching, i.e., on geometry, 12 students (54.5%) were sure they did, 6 students (27.3%) answered somehow yes, 3 students (13.6%) said no, and 1 student (4.6%) did not know. The large majority of the students (18 out of 22) thought that their results in school could be improved with MatES.

We asked the students if they used MatES at home, supposing that they had a computer of their own (all except 2 have their own computer at home). 11 students (50%) answered “certainly”, the other 11 students answered “somehow yes”, and none answered “no” or “somehow no”. This extremely positive result shows that the students may be convinced of the benefits of MatES. But they may also have given this answer to value their teacher’s efforts in this experiment.

No real correlation can be found in the answers given to the question if they could imagine learning without a teacher. 4 students (18.1%) are convinced they could, 10 students (45.5%) said that they still somehow need a teacher, and 8 students (36.4%) answered that they still need a teacher.

Finally, they were asked if they enjoyed working with MatES. 11 students (50%) said “yes”, 9 students (40.9%) said “yes, a lot”, and 2 students (9.1%) said “somehow yes”. None of the students disliked working with MatES. This motivation doing mathematics might be one explanation for the better school results.



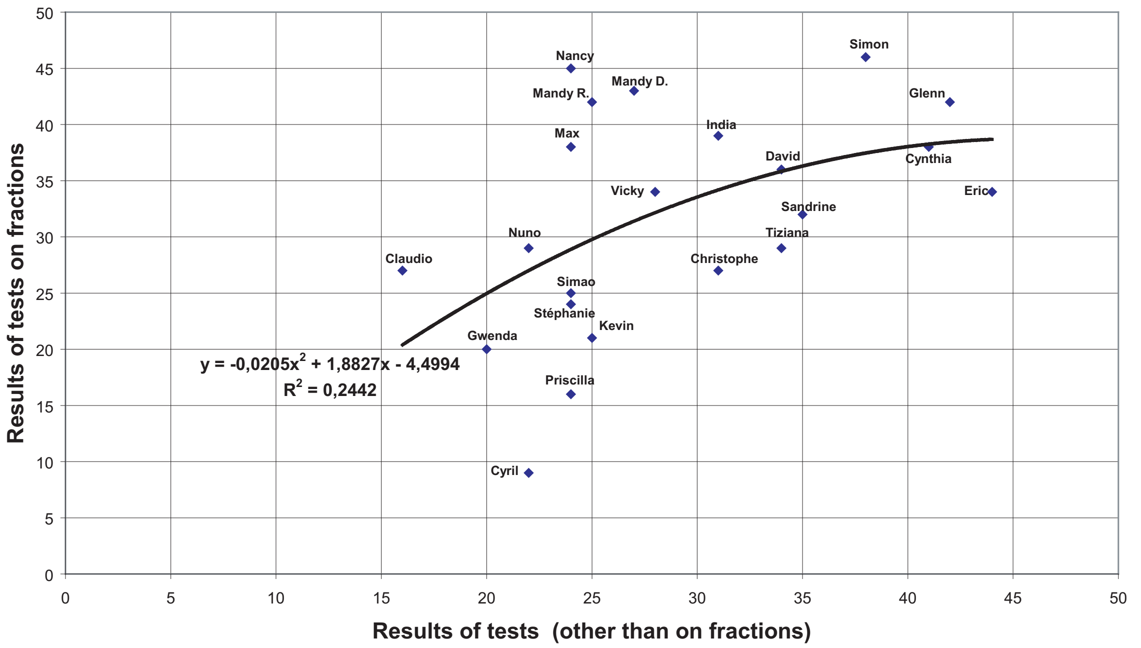


Figure 12.4: Average of the tests on geometry (x-axis) and average of the tests on fractions (y-axis) with (1) the regression line and (2) a polynomial adjustment.

12.3.2.2 Comments Concerning MatES

Generally, MatES returned only very few results, normally one, rarely more than 3. We asked the students to give their opinion about the number of results. None of them stated that there were too few, 1 answered that there were too many, and 21 (out of 22) said that there were neither too many nor too few results.

The next question was about the quality of the search results yielded by MatES. We asked if they found the answer to their question in the results yielded by MatES. Nobody said "never" or "rarely", 1 student (4.6%) said "in half of the cases", 16 students (72.7%) said "most of the time", and 5 students (22.7%) answered "always".

One important question was about the constraint to enter complete questions. No student said that this was awkward, 7 students (31.8%) answered that they accept entering a complete question but that they did not like it, and 15 students (68.2%) answered that this was no problem at all.

We asked the students what they especially liked about MatES. 2 students (out of 22) had nothing to say. The comment most mentioned was about the quality of the search engine (10 comments): MatES always yields an answer, answers are always right and respond to the question, etc. A second comment that was often made dealt with the explanations and the content of the clips (9 comments): well explained, one understands the subject better, etc. Other comments were that MatES has a lot of knowledge (3 comments), that the student must not ask the teacher, and one must not wait for the teacher to be available (3 comments), that an explanation can be watched several times (3 comments), that the answers are short (1 comment), that the illustrations and explanations are presented nicely (1 comment), that one can use computers (1 comment), and that students give the explanations in the clips (1 comment).

We asked the students what they especially disliked about MatES. 7 students (out of 22) had nothing to say. The comment mentioned most was about technical problems (7 comments): the computer or MatES got stuck. Another comment that was often made dealt with the interaction with MatES (4 comments): that MatES only tolerates little errors in the questions, that one has to enter complete questions, and that it is easier to communicate with a teacher. Other comments were that the questions must be formulated in French[2] (2 comments), that the video of the presenter is disturbing (1 comment), that the system sometimes returns bad answers (1 comment), and that some explanations are too complicated (1 comment). Two students complained that it lasts too long to watch a clip to get the answer to one's questions, and that asking a real teacher or a friend would be simpler and faster.

12.3.3 Analysis of the Log-files

The log-files show that nearly all queries were well formulated. Only very little "out-of-the-topic" questions were voluntarily placed. In average, each student entered 8.5 questions per lesson (50 minutes). An average of 4 questions was asked in the lesson with the lowest number of questions, and an average of 17 questions was asked in the lesson with the highest number of questions. There was no difference concerning the number of submitted questions between an ordinary lesson and one in which they wrote a test. There was no student who entered exceptionally many, or exceptionally few questions. Weak or strong students asked approximately the same number of questions, independently of whether it was a normal lesson, or a test.

An interesting observation was that students sometimes entered the same or a previous question again. This observation is well documented in literature on surveys about how students search on the Web [FDD$^+$99, HS00]. Students tend to re-enter a previous question (or keywords) where they

[2] In Luxembourg the mathematics course is held in French, which is the second foreign language.

are sure to get a result. They often return to ”landmarks” where they received good answers. We can observe that some students re-entered 3 or 4 times the same question in one lesson. This observation has already been made in the former experiment with CHESt (see section 11.5).

12.3.4 General Observations

In week 1 of the experiment, the students were astonished about the way they were able to learn fractions; that there were no classical “theoretical” lessons, and that they had to use a computer tool to ask questions in order to acquire new knowledge on their own. The fact that they had to enter complete questions was a problem during the first lessons. Firstly, entering so many words was a burden for most of the students because it demanded greater effort, and because they were accustomed to keyword based search engines (e.g., Google). Secondly, at their school level, they did not yet learn how to formulate questions. Thirdly, some errors in the tool involved that MatES blocked frequently. This caused frustration, because they had to type their question again.

After the second week, all students became accustomed to this kind of teaching. Entering complete questions became generally accepted. We witnessed that most of the students entered questions very quickly. It seemed that they had a lot of experience typing on a computer (possibly by chatting on the Internet). The students also found out that formulating questions was not so difficult, because in most of the cases the instructions of the exercises were already close to the form of a question; e.g., the instruction, “Simplify the following fractions” could become the question, “How do you simplify a fraction?”.

With the progression of the experiment, the students became more and more amazed by this kind of learning mathematics. Some expressed that the exercises were fun, others enjoyed the video sequences and started to know by heart the names of presenters. We observed that students remembered the clips by some kind of characteristic, e.g., a presenter that pronounces a certain word badly, a nice illustration inside a clip, or a presenter who explains very well. It was interesting to see that such characteristics were very useful for the students. Thus, if someone had to search for a clip about the simplification of a fraction, (s)he said: “That’s Lynn’s clip”, or “That’s the clip with the pizza-example”.

We were impressed by the very positive atmosphere in the classroom. Every student was occupied with her/his own exercises, and could progress in her/his own rhythm. Some worked quite fast, others were slower. All the students used headsets. It was pleasantly calm in the room. Students were allowed to communicate (except during the 2 tests). Most chats resembled these comments: “What clip did you find for this exercise?”, “Did you get an answer for this exercise?”, “Have you already finished this exercise?”, etc.

What was unexpected was that the students asked for permanent assistance from the teacher. In the first week, most of the questions were about how to use MatES, or about technical problems, i.e., the computer or MatES blocked. As for the other weeks, the kind of questions changed rapidly in a more mathematical context. Questions resembled these ones: “I don’t understand how to solve this exercise, please help me!”, “Last year we did not do it like that.”, “Is it OK if I write it in this way?”, etc. In fact, students were not used to working individually and autonomously, especially not in a mathematics course. Hence, students were often unsure if they understood correctly, or if their solution was right.

At the end of week 5, students were sad that the experiment was over, and that they returned to a “classical” kind of learning. Several students asked to get a copy of MatES for their personal use at home.

12.4 Discussions and Conclusions

In this section, we will analyze the data from the experiment (section 12.3), and will try to figure out if the better results can be traced back to the use of MatES, or if there are other reasons.

12.4.1 Reasons Other than MatES

Was the subject of fractions easier to understand than the subject of geometry? Different teachers confirmed that both subjects have a similar level of difficulty the way they are taught in school.

Did the students have any knowledge about fractions? All students already had some basic knowledge on fractions, but also on geometry, because more or less the same efforts were spent on teaching these subjects over the previous 3 years.

Were the tests about fractions easier? The tests were similar, even identical to those made in the other classes the same, or a previous year. Furthermore, all tests (about geometry and about fractions) were corrected by two teachers.

12.4.2 Better Understanding

Do the explanations from MatES help the students to better understand the explanations from classical sources (e.g., books, notes on the blackboard or verbal explanations from the teacher)? Nine students stated that the explanations from MatES were very good and 3 students stated that MatES has a lot of knowledge. Nearly all students (21) stated that they found the right information using MatES, and 18 stated that they learned better with MatES. Here are some explanations:

- The semantic search engine helps the students to find a good answer quickly; in other words, they do not have to wait for the teacher to ask their question.
- The answer is very precise and short, unlike in a book, or unlike the long explanations from the teacher.
- The explanations are simple, and straightforward.
- The student can navigate through a clip, and stop at a more important part, or watch a clip several times.
- The information is presented in a more appealing form than in a book or on the blackboard. For example, several students remembered a certain information because they remembered a certain characteristic in a clip.
- The multimedia aspect activates more senses. The student hears, reads and sees the same information.
- Illustrations are used to explain a certain topic, which is more expressive than verbal communication [MG90, MS94].
- The video sequences show the presenter on the blackboard (or whiteboard). This is the students' common view in a classroom, and should create a kind of virtual classroom atmosphere; it is supposed to be serious work, and not a game.
- The motion on the screen keeps the students concentrated on what they do and draws their attention to what the presenter is explaining.

- The presenters were students. Some students accept more easily to be taught by colleagues than by adults.
- Students quickly acquired the specific vocabulary about fractions. If an unknown expression was used in a clip, then they could simply ask MatES to explain it.

12.4.3 Higher Motivation

Every teacher knows how pleasant it is to teach in a class with motivated students. Industrious students generally produce good results, because they are willing to put more time and effort into learning. However, many students do not have an innate motivation to learn. Therefore, it is necessary to find means to convince them of the importance, and the need for this training.

There is the *intrinsic motivation* that is related to MatES, and the *extrinsic motivation* that is not related to MatES. The intrinsic motivation originates particularly in the desire of the students to understand the explanation of the presenter in the clip, to master this matter as well as the presenter, and to correct it (e.g., several students claimed that some subjects were not well presented, and that they could do better). It is also important to call forth the extrinsic motivation of the students during the experiment in order for every student to put all her/his effort in working correctly with MatES, and not to spend too much time watching and enjoying the clips. An extrinsic motivation was that the students who finished their exercises in school would have less homework.

The higher motivation can be traced back to MatES, because neither geometry, nor fractions are de facto motivating for most of the students. Maybe students have a small preference for geometry because they can use instruments, make drawings, etc. whereas fractions are pure calculations. However, 20 students (90.9%) asserted that they enjoyed working with MatES. We even heard the statement: "With this [MatES], even mathematics is fun". Here are some reasons for their increased motivation:

- The use of new technologies is in general motivating for students. A similar conclusion is drawn from [PBR06], where students use tablet PCs, and from [HPM05], where a game-like tool was used to stimulate learning by making unpopular subjects fun.
- Everything that is different from the normal kind of teaching is, at least at the beginning, motivating for the students. Thus, the lessons took place in a computer room, and they used a computer tool in mathematics, which is unusual.
- The explanations are presented in a more appealing form, i.e., short multimedia clips (see section 12.4.2).
- The student has the impression that (s)he is in control of the lesson. There is no teacher who dictates what to do next.
- The student is active in her/his learning process. Everyone is able to do something constantly, and to build her/his knowledge by his own action.
- In traditional courses, the stronger students mostly perform better, which is frustrating for others. However, this style of teaching easily enables each student to progress according to her/his own capacities, and none is embarrassed.
- The lessons can be perceived as a kind of adventure where the student plays the role of an explorer who discovers new information.

- The student understands better (see section 12.4.2), and therefore has no reason to resign. In contrary, (s)he realizes that mathematics are finally not so difficult at all.

12.4.4 Greater Efforts

In our experiment, the students had to do a lot of extra work. These greater efforts could explain the better results. Firstly, as there were no "theoretical" lessons, the students spent most of the time doing exercises. Therefore, they had more time to find out about their weaknesses, to complete their knowledge, and to test it by solving the exercises. Secondly, they were aware that they had to learn and understand the theory in order to complete the exercises. Thus, it was in their own interest to acquire the necessary theory as soon as possible. Thirdly, the students knew that they had to finish their exercises at home. Hence, it was in their own interest to work in an efficient way in school in order to minimize their homework. Fourthly, weaker students had more homework because they progressed at a slower rhythm. This supplementary work and the required efforts to do it could have helped them to become better.

12.4.5 Different Pedagogy

In a classical mathematics course, the student receives information from the teacher, and has to understand and to memorize it. The volume of information and the velocity at which the information arrives often overwhelm weak students [WMSB01]. Furthermore, if the student is not convinced of the importance of the information and the training, then the lesson is not effective.

MatES proposes a completely different pedagogical approach, which fosters autonomous and exploratory learning, and where the student becomes active in the learning process. With MatES, the student receives information only when (s)he asks for it. In this approach, the student directs her/his training; what (s)he wants to read, what is the rhythm of progression, how often (s)he wants to read the same information, etc. She/He does not depend on the teacher or on the other students. Therefore, a weak student can progress in her/his appropriate rhythm. She/He can acquire knowledge about the same concepts as the rest of the class. If (s)he is a strong and ambitious student, then (s)he can progress faster and do more advanced exercises than required. She/He does not have to stay silent and inactive during the time the teacher explains the same information again to weaker students.

However, let us notice that this style of teaching does not foster learning by heart compared to *intelligent learning*. We observed that some weak students, who had acceptable results in mathematics in the past since they could study by heart, had worse results with MatES. To learn by heart is a strategy adopted by the students from secondary education which is not always effective.

12.4.6 Outcomes and Lessons Learned

The data from the tests show that the students had better results when they used MatES. However, it cannot be proven if these improvements were really the direct consequence of the use of MatES. It is a fact that working with MatES was more motivating for the students, which in turn had a positive influence on the students' learning and understanding. Therefore, MatES indirectly contributed to improving the students' school results.

An open question is how long students remain motivated with MatES, because today students quickly lose interest in what they do. Although students enjoyed using MatES for 5 weeks, the tool could become as boring as any ordinary schoolbook after another 5 weeks.

A regrettable fact is that students perceive computers in general and software in particular as a toy. The teacher's first task is to convince the students that the computer or the software is not a toy but a helpful tool. Games are funny at the beginning, but the student loses interest very quickly and asks for new things. However, if the student is convinced of the advantages of such a tool, then (s)he is likely to continue using it. For example, if students have to write a report, they immediately ask to use a word processor; thus they could write it without using a computer. A word processor is not perceived as a game, but as a useful tool.

We learned that students do only successfully and correctly use a computer tool, if they are convinced of its benefits, and if they know how to use it correctly. An example is a conversation between two students that we recorded during a lesson. Both students talked about the problem as they had to solve an exercise. Then, one yelled spontaneously: “Let’s ask MatES!”. They knew that there was no obligation to use MatES, but they were aware that it could be of some help.

The success of our experiment is also partially due to the fact that the students were guided during the whole experiment, which is a requirement for a successful computer based training [Mar03, FDD$^+$99, NPSR99, CMZ03]. Therefore, MatES did not reduce the volume of work for the teacher [Ows97]. It is clear that students need more guidance and ask more questions if they become active in their learning process. Furthermore, in traditional learning environments, teachers are primarily responsible for the organization, delivery and assessment of content acquisition by students in their courses. This changes as soon as teachers use e-Learning technologies. They receive additional roles like technology specialist or administrative advisor.

The quality of the semantic search engine is a crucial factor of the success of MatES. We know that students generally dislike getting multiple results; they do not even consider them all [FDD$^+$99]. Students have clear expectations about the requested search result. Even if MatES yielded 5 results, which was quite unusual because normally the number of results was smaller than 3, some students complained about this “high” number of clips (“Do I need to watch all of them?”).

Part IV

Conclusion and Future Work

Chapter 13

Conclusion and Key Results

This thesis is about the elaboration and study of an E-Librarian Service; a tool that accepts queries in natural language, and returns few but semantically pertinent answers in form of short multimedia clips. It was tested in an educational environment, as a kind of "virtual teacher" who understands the students' questions. Here is a summary of the main contributions and key results of this research work.

- We developed an algorithm based on partial parsing and ontology technology, which translates a natural language query into a logical form, i.e., a Description Logics-concept description. We also elaborated a solution to resolve ambiguities in the natural language sentence.
- We developed an algorithm based on non-standard inferences in Description Logics — a modified version of the concept covering problem — and ontology technology to find and retrieve semantically relevant resources in a multimedia knowledge base. We also elaborated a solution to quantify the semantic distance between a query and a set of matching documents in order to identify the most relevant answer. Our solution significantly improves domain search engines.
- We empirically showed in benchmark tests the quality and the reliability of our E-Librarian Service. In the majority of the cases, it delivered the correct answer as best match. Also, it yielded very few answers only — normally just one answer, a few times more than three — and always at least one answer.
- This thesis is fully in the stream of current Semantic Web thinking. We used technologies like RDF and OWL to build a semantic search engine that can easily be adapted to other domains.
- Current ontology research aims to provide short but appropriate ontologies for all imaginable domains. We created three different ontologies; two rather small ones about computer history and fractions in mathematics, and one relatively large one about networks in computer science and Internetworking. The ontologies are available on demand.
- Three multimedia knowledge bases were produced, each containing lots of pedagogically rich material about computer history, fractions in mathematics, and networks in computer science and Internetworking. They are available on demand.
- We developed different prototypes that can very easily be used. Some prototypes are available as "standalone application" which do not require any installation or configuration procedure,

and can immediately be started from CD/DVD. Some prototypes can be used via a Web page. The most recent version of our E-Librarian Service has a Web service interface. In this way, developers can build their own applications that use our semantic search engine. The prototypes are available on demand.

- The architecture of our E-Librarian Service allows a distributed approach: the different layers (e.g., semantic search engine, or knowledge repositories) can be located on the local machine or somewhere on the Internet.

- Experiments in an educational environment showed that students perceive our E-Librarian Service as a useful tool and not as a game, and accept to enter complete questions instead of keywords: 22% of the users would prefer to enter complete questions instead of keywords, 69% would prefer to enter complete questions instead of keywords only if this would yield better results and 8% would dislike this option.

- One of the main outcomes of this thesis is that we showed empirically that school results can be improved when students use our E-Librarian Service. Although these results were achieved in a very precise domain and are representative for one particular class only, it is one of the rare scientifically documented case studies of the benefits of an E-Learning tool. We measured an overall improvement of 5% in the students' results compared to their past results. 50% of the students improved their school results, 41% of them progressed very much. One of the main reasons for these positive results is that the students were more motivated, and therefore willing to put more effort into learning and acquiring new knowledge.

Chapter 14

Thesis Summary

Although educational content in electronic form is increasing dramatically, its usage in an educational environment is poor, mainly due to the fact that there is too much of (unreliable) redundant, and not relevant information. Finding appropriate answers is a rather difficult task being reliant on the user filtering of the pertinent information from the *noise*. Turning knowledge bases like the online tele-TASK archive into useful educational resources requires identifying correct, reliable, and "machine-understandable" information, as well as developing simple but efficient search tools with the ability to reason over this information.

Our vision is to create an *E-Librarian Service*, which is able to retrieve multimedia resources from a knowledge base in a more efficient way than by browsing through an index, or by using a simple keyword search. Our premise is that more pertinent results would be retrieved if the *search engine* understood the sense of the user's query. The returned results are then logical consequences of an inference rather than of keyword matchings. Our E-Librarian Service does not return the answer to the user's question, but it retrieves the most pertinent document(s), in which the user finds the answer to his/her question.

Our E-Librarian Service is an ontology driven expert system about a given domain: computer history, fractions in mathematics, and networks in computer science. It relies on specialized and hierarchically organized knowledge base, and specific reasoning services. The documents in the knowledge base are described by additional data — called metadata — that are encoded using a specific framework; we use the W3C recommendation *Web Ontology Language* (OWL) and its sublangauge OWL DL that relies on Description Logics (DLs).

DLs are a family of knowledge representation formalisms that allow to represent the knowledge of an application domain in a structured way, and to reason about this knowledge. In DLs, the conceptual knowledge of an application domain is represented in terms of *concepts* (unary predicates) and *roles* (binary predicates). Concepts denote sets of individuals, and roles denote binary relations between individuals. Complex descriptions are built inductively using concept constructors, which rely on basic concept and role names. Different DL languages distinguish themselves by the kinds of constructs they allow. In particular, we use the language $\mathcal{EL}$ that has structural subsumption and allows conjunction ($\sqcap$), existential restriction ($\exists r.C$), and the top concept ($\top$).

In our E-Librarian Service, the user can enter his question in a very simple and human way; in natural language (NL). Linguistic relations within the user's NL question and a given context, i.e., an ontology are used for the semantic interpretation into a DL-concept description. This is achieved by mapping the canonical form of the words from the user question to ontology concepts, and by transforming them into a conjunctive formula. Finally, the roles' arguments are resolved, and the DL-concept description is put into a normal form.

We conceived a solution to resolve ambiguities in the NL input, e.g., multiple sense words. Let us consider as an illustration the word "Ada", which can refer to the programming language named "Ada", but it can also be the name of the person "Augusta Ada Lovelace". The correct interpretation can only be retrieved accurately by putting the ambiguous word in the context of a complete question. Thus, the context of the sentences "Who invented Ada?" and "Did the firms Bull and Honeywell create Ada?" reveals that here "Ada" is the programming language, and not the person "Ada". Hence, our NLP-module is able to correctly translate the question "Who invented Ada?" into the DL-concept description:

$$Q \equiv \mathsf{Ada} \sqcap \exists \mathsf{wasInventedBy.Person}.$$

The retrieval of semantically pertinent documents is based on non-standard inferences in DLs; the *least common subsumer*, the *semantic difference*, the *concept covering problem*, and the computation of the *semantic distance*. Among all the documents that have some common information with the user query, our E-Librarian Service identifies the most pertinent match(es), keeping in mind that the user expects an exhaustive answer while preferring a concise answer with only little or no information overhead. Also, our E-Librarian Service always proposes a solution to the user, even if the system concludes that there is no exhaustive answer. By quantifying the missing and supplementary information, the system is able to compute, and visualize the quality and pertinence of the yielded documents.

In more details, we define a *cover* as being a DL-concept description C (candidate document) that shares some information with another DL-concept description Q (query). The *best cover* is defined as based on the remaining information in the query (*Miss*) and in the cover (*Rest*). The Miss is the part of the query that is not part of the cover, and the Rest is the information that is part of the cover but not required by the query. The best cover can be assumed to be the cover with the smallest Miss and Rest, with a preference to a minimized Miss. Our E-Librarian Service aims to give an exhaustive answer in the first place, i.e., to yield an answer that covers the user's query as much as possible, even if there is more information in the answer than required. In the second place, the Rest is considered in order to rank the results with equal Miss. To provide an illustration, let us suppose that there are 5 documents in the knowledge base with the following semantic descriptions (terminology):

$$\begin{array}{l}
\mathsf{LO}_1 \equiv \mathsf{Protocol} \sqcap \mathsf{Communication}\\
\mathsf{LO}_2 \equiv \exists \mathsf{howWorks} \sqcap \mathsf{TCPIP} \sqcap \mathsf{Protocol} \sqcap \mathsf{Communication}\\
\mathsf{LO}_3 \equiv \mathsf{Protocol} \sqcap \mathsf{Communication} \sqcap \exists \mathsf{hasTask}.(\mathsf{ErrorHandling} \sqcap \mathsf{ProtocolService} \sqcap \mathsf{Service})\\
\mathsf{LO}_4 \equiv \mathsf{Protocol} \sqcap \mathsf{Communication} \sqcap \exists \mathsf{hasTask}.(\mathsf{FlowControl} \sqcap \mathsf{ProtocolService} \sqcap \mathsf{Service})\\
\mathsf{LO}_5 \equiv \mathsf{FlowControl} \sqcap \mathsf{ProtocolService} \sqcap \mathsf{Service}
\end{array}$$

Let us suppose that the user has entered the NL question "What are the tasks of TCP/IP?", and that the question was translated into the following $\mathcal{EL}$-concept description: $Q \equiv \mathsf{TCPIP} \sqcap \exists \mathsf{hasTask}$. The aim is now to identify the documents within the terminology that cover the query, i.e., that have something in common with Q; these are: LO_1, LO_2, LO_3, and LO_4. The best cover is the one with minimal Miss and Rest, with a preference to the minimal Miss.

	size of the Miss	size of the Rest
LO_1	$\lvert\mathsf{TCPIP} \sqcap \exists\mathsf{hasTask}\rvert = 3$	$\lvert\top\rvert = 0$
LO_2	$\lvert\exists\mathsf{hasTask}\rvert = 2$	$\lvert\exists\mathsf{howWorks}\rvert = 2$
LO_3	$\lvert\mathsf{TCPIP}\rvert = 1$	$\lvert\mathsf{ErrorHandling} \sqcap \mathsf{ProtocolService} \sqcap \mathsf{Service}\rvert = 3$
LO_4	$\lvert\mathsf{TCPIP}\rvert = 1$	$\lvert\mathsf{FlowControl} \sqcap \mathsf{ProtocolService} \sqcap \mathsf{Service}\rvert = 3$

LO_3 and LO_4 are the best covers, and are delivered as an answer to the user's query. Both documents have the same Miss and Rest, so that their rank is the same. It is interesting to mention that the concept TCPIP does not appear in one of the best covers, although it appears in the query and in LO_1. This shows that the best cover is not computed on a statistical evaluation of keywords, but that it is in fact the result of a logical inference.

In benchmark tests, our E-Librarian Service was compared to a keyword-based search engine. For the evaluation, we selected a lecture on Internetworking from the online tele-TASK archive[1], and split it into 40 smaller learning objects (LOs). A set of 123 NL questions has been created, and we indicated for each question the relevant answer(s) to be delivered. An answer from our E-Librarian Service is called a *perfect hit* if it covers the query completely (Miss = Rest = 0), and a *sufficient hit* if it covers the query completely but contains more information than necessary (Miss = 0, Rest$>$ 0). The results achieved with our E-Librarian Service have been compared to the results of a traditional keyword-based search engine. The outcome is that our E-Librarian Service scored better than the keyword search regarding the pertinence of the results.

	perfect hits	sufficient hits	total queries
E-Librarian Service	93 (76%)	112 (91%)	123 (100%)
Keyword-based search	9 (7%)	103 (84%)	123 (100%)

The precision of our E-Librarian Service is confirmed by the fact that in average less than 0.7 LOs are delivered in addition to the perfect answer (compared to 6 LOs for the keyword-based search). Furthermore, our E-Librarian Service usually achieves the correct answer with no additional information (93 out of 123), and in a few cases one (12 out of 123) or two (6 out of 123) supplementary LOs. The keyword-based search in general delivers much more secondary LOs.

Our E-Librarian Service was implemented prototypically in three different educational tools. A first prototype is CHESt (*Computer History Expert System*); it has a knowledge base with 300 multimedia *clips* that cover the main events in computer history. A second prototype is MatES (*Mathematics Expert System*); it has a knowledge base with 115 clips that cover the topic of fractions in mathematics for secondary school w.r.t. the official school programme. All clips were recorded mainly by pupils. The third and most advanced prototype is the "Lecture Butler's E-Librarain Service"; it has a Web service interface to respect a service oriented architecture (SOA), and was developed in the context of the Web-University project at the Hasso-Plattner-Institute (HPI).

Two major experiments in an educational environment — at the Lycée Technique Esch/Alzette in Luxembourg — were made to test the pertinence and reliability of our E-Librarian Service as a complement to traditional courses. The first experiment (in 2005) was made with CHESt in different classes, and covered a single lesson. The second experiment (in 2006) covered a period of 6 weeks of intensive use of MatES in one class. There was no classical mathematics lesson where the teacher gave explanations, but the students had to learn in an autonomous and exploratory way. They had to ask questions to the E-Librarian Service just the way they would if there was a human teacher.

In both experiments we asked the students about their liking of such a tool and their acceptance to enter complete questions instead of keywords. In the first experiment, 22% of the students answered that they would have no problem to enter complete questions instead of keywords, 69% preferred to enter complete questions instead of keywords if this yielded better results, and 8% disliked this option. In the second experiment, no student stated that this was awkward, 31.8% answered that they accepted having to enter a complete question but that they did not like it, and 68.2% answered that this was no problem at all.

[1] `http://www.tele-task.de/page42_mode1_series599.html`

The major outcome of the second experiment is that school results can be improved when students use our E-Librarian Service. Although these results were achieved in a very precise domain, and are valid for one particular class only, it is however one of the rare scientifically documented case studies of the benefits of an E-Learning tool. We measured an overall improvement of 5% in the students' results compared to their past results. 50% of the students improved their school results, 41% of them progressed very much. One of the main reasons for these positive results is that the students were more motivated, and therefore willing to put more effort into learning and acquiring new knowledge.

Chapter 15

Future Work

Contents

This chapter will present some possible improvements of our E-Librarian Service. Section 15.1 will give an overview of some future work, while sections 15.2 and 15.3 will describe two topics in more details: the automatic generation of metadata, and the user feedback, respectively.

15.1 Overview of Possible Improvements

Although we presented a running and evaluated system in this thesis, there is still space for improvement. Here is an uncomplete list of some future work:

- The scalability aspect of our E-Librarian Service should be studied. The question is: will the quality of the semantic search engine remain as reliable as presented in chapter 8, when the knowledge base contains ten, hundred, or thousand times more documents?
- Experiments in an educational environment showed that our E-Librarian Service is a useful and efficient E-Learning tool. One key question has already been mentioned in chapter 12 and should be studied in more detail: will our E-Librarian Service remain as "attractive" to the students when it it used over a longer period of time? The duration of the longest experiment was 5 weeks.
- Our E-Librarian Service relies on the quality of the metadata that are used to describe the documents in the knowledge base. As stated in section 5.3, we supposed that a consistent semantic annotation existed. However, the creation of metadata is an awkward task that should eventually be done in a (semi)automatic process. We briefly explored one solution that will be detailed in section 15.2, but which currently gives no reliable results.
- It should also be explored in how far user query prediction [MZ07] or user feedback could improve the quality of our E-Librarian Service. We present some theoretical thoughts about that topic in section 15.3.

15.2 Automatic Generation of Metadata

The creation of semantic annotation is and should neither be the task of the user, nor of the creator of the document. The user (e.g., a student) and the creator (e.g., a lecturer) are not necessarily computer science experts who know how to create metadata in a specific formalism like XML, RDF, or OWL. Furthermore, the creation of metadata is a subjective task and should be done conscientiously. The automatic generation of reliable metadata is still a very difficult problem, and currently a hot topic in the Web 2.0 movement [KPT+04, Tro03, JNL+05]. Different solutions exist, e.g., [WPS+04, BBT+06], but cannot offer a universal and reliable infrastructure.

In this section, we will briefly present a solution that we explored to generate semantic annotations for university lectures. It is based on the extraction of keywords from two data sources (audio and slides), which are then mapped to ontology concepts. The quality of our solution is evaluated via different benchmark tests and was published in [RLM07].

15.2.1 Extraction Method

The input of our algorithm is the imperfect transliteration from a speech recognition engine, and the textual content from the slides. Using speech recognition to annotate videos is a widely used method [HKW02, CH03, YOA03, HK05, NWP03, SW06c, SW06a, ZZ03, WNP06]. Keywords from both sources are then mapped to ontology concepts. Only the relevant keywords according to a given rank are considered as metadata.

The ranking algorithm works as follows. For each identified concept, we compute its hit-rate h, i.e., its frequency of occurrence inside the document. Only the concepts with the maximum hit-rate compared to the hit-rate in the other documents are used as metadata. For instance, the concept

Topology has the following hit-rate for the five documents LO_1 to LO_5:

	LO_1	LO_2	LO_3	LO_4	LO_5
h	0	4	3	7	2

This means that the concept Topology was not mentioned in LO_1 but 4 times in LO_2, 3 times in LO_3, etc. For a given rank, e.g., $d = 1$ the concept Topology is relevant only in the document LO_4 because it has the highest hit-rate. For $d = 2$ the concept is associated to the documents LO_4 and LO_2, i.e., the two documents with the highest hit-rate. We empirically found out that the best results were with $d = 2$.

15.2.2 Evaluation and Discussion

The quality of the automatically generated metadata was tested in a benchmark test with our E-Librarian Service. We used the same test conditions as for the benchmark test described in section 8.3, but instead of our manually created semantic annotation we used the automatically generated metadata.

A first outcome is that using audio data in addition to the content of the slides does not improve the search results. This outcome must be due to the fact that the annotations for both, audio and slides, are semantically very close, so that their combination introduced no more additional information.

A second outcome is the fact that with a harder retrieval constraint by considering a ranking ratio, the recall decreases but the precision increases. In that case, we have a greater value for the perfect hits[1] as well as a small average Rest, but also a smaller recall w.r.t. sufficient hits.

Finally, the semantic search generally has worse results when considering the general recall w.r.t. sufficient hits, but double, even triple the precision w.r.t. perfect hits and average rest. It would seem that the quality of our automatically generated annotations is neither good nor precise enough for the semantic search engine to significantly improve its search results.

15.2.3 Discussion and Improvements

The quality of the generated metadata is not sufficient to be used efficiently by a semantic search engine. We think that the weak quality of the generated metadata has two main reasons.

The first reason is the quality of the audio data processing, and in particular the quality of the speech recognition, e.g., inappropriate acoustics in the room, bad microphone, and no training of the system.

Secondly, natural language in general is often a source of linguistic ambiguities. We had situations where the speech recognizer created ambiguities like the German word “Mann”, which can mean “a man” (German: “Mann”) but it can also mean “one” (German: “man”), or a network “MAN” (Metropolitan Area Network). All three are pronounced in the same way.

We suggest two major improvements. First, by comparing the synchronized audio data with the data from the slides one can find overlapping areas. For example, the lecturer speaks in a certain part of the presentation about “host ID” and shows a slide with the word “host ID”. Then, it is obvious that the word “host ID” is a relevant word in the context of this document, i.e., more important than a word that was only found in one of both data sources (audio or slide).

[1] An answer from our E-Librarian Service is called a *perfect hit* if it covers the query completely (Miss = Rest = 0), and a *sufficient hit* if it covers the query completely but contains more information than necessary (Miss = 0, Rest$>$ 0). See section 7.4.2 for more details.

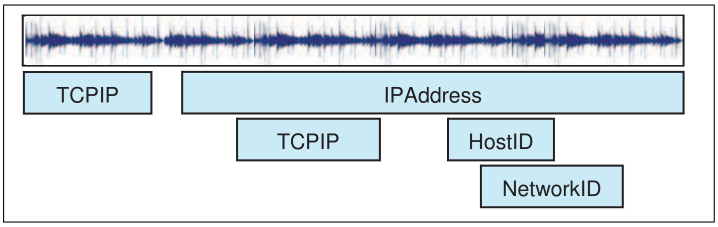


Figure 15.1: Example of 4 identified chains inside a lecture part about IP addressing.

Secondly, documents can be divided into cohesive areas (*chains*) of accumulated appearance of an equal word. For example, in the space of 5 minutes the speaker uses the expression "host ID" 3 times: this segment of 5 minutes is called a chain about the concept HostID. A chain is always about one specific word. Chains overlap when the speaker uses different relevant words several times during the same time interval (see figure 15.1). The resulting granularity of the segmentation depends on the allowed gap (threshold) between two identical words.

15.3 Improving Search Result Quality with User Feedback

In this section, we will briefly present some ideas how to improve the quality of our E-Librarian Service by using the user's intellectual capabilities. These are: direct user feedback, collaborative tagging and social networks, and diversification of user feedback.

15.3.1 Direct User Feedback

Direct user feedback can be achieved in different forms. The most simple way is to let the user determine whether a given result set of documents really is appropriate according to his/her question or not. The E-Librarian Service has to keep track of user feedback and to channel that data into the rank computation of the document result sets.

The E-Librarian Service faces the problem to provide both an *objective answer*, as well as a feedback-driven and therefore more or less *subjective answer*. Therefore, it displays both the (objective) best covers and the (subjective) feedback-based results. Thus, the user has the possibility to see objectively computed results, and the results according to the opinion of other users. If both results fit in the way that they both display the same top-rank result, the quality of our algorithm is confirmed.

15.3.2 Collaborative Tagging and Social Networks

User generated keywords (tags) are an additional source for the semantic annotation of documents in a knowledge base. A user might provide additional, otherwise not available semantic annotation. In this regard, *collaborative tagging* has gained increasing popularity, which is demonstrated by the growing number of prominent tagging and annotation sites such as Delicious[2], Flickr[3], or Bibsonomy[4].

An additional source of information is provided by the *social networking* information of the tagging service. Based on this networking information a similarity measure for documents can

[2] `http://www.del.icio.us/`
[3] `http;//www.flickr.com/`
[4] `http://http://www.bibsonomy.org/`

be determined. Users, who tag the same documents with the same or similar keywords can be considered to have similar or common interests. By retrieving documents with similar tags, similar documents can be determined.

15.3.3 Considering the User's Expectations

Different users asking the same question might expect different answers. This is due to the fact that different users prefer different levels of complexity, of difficulty, and of elaborateness [LTV03, Mor05]. Moreover, different users come from different backgrounds, have different motivations, and thus, a different context. The user must be given the means to specify, if (s)he prefers complex and precise documents, or if a short overview about the requested topic is sufficient.

If our E-Librarian Service keeps track of the user's actions, then statistics can be gathered about document usage. If a user has already accessed and used a given document, this information can be used to customize the computation of the best cover w.r.t. the previous knowledge of the user.

Bibliography

[All94] James Allen. *Natural Language Understanding.* Addison Wesley, 1994.

[AMO+03] Jürgen Angele, Eddie Mönch, Henrik Oppermann, Steffen Staab, and Dirk Wenke. Ontology-based query and answering in chemistry: Ontonova @ project halo. In *International Semantic Web Conference (ISWC)*, pages 913–928, 2003.

[ART95] Ion Androutsopoulos, Graeme D. Ritchie, and Peter Thanisch. Natural language interfaces to databases – an introduction. *Language Engineering*, 1(1):29–81, 1995.

[AvH04] Grigoris Antoniou and Frank van Harmelen. *A Semantic Web Primer.* The MIT Press, 2004.

[Bau96] Rüdeger Baumann. *Didaktik der Informatik.* Klett, Stuttgart, 1996.

[BB04] Helene Blowers and Robin Bryan. *Weaving a library Web: A guide to developing children's Websites.* American Library Association, Chicago, 2004.

[BBT+06] Marco Bertini, Alberto Del Bimbo, Carlo Torniai, Rita Cucchiara, and Costantino Grana. Mom: Multimedia ontology manager. A framework for automatic annotation and semantic retrieval of video sequences. In *ACM SIGMM*, pages 787–788, 2006.

[BBZ+05] Mark H. Burstein, Christoph Bussler, Michal Zaremba, Timothy W. Finin, Michael N. Huhns, Massimo Paolucci, Amit P. Sheth, and Stuart K. Williams. A semantic web services architecture. *IEEE Internet Computing*, 9(5):72–81, 2005.

[BCFB04] Dario Bonino, Fulvio Corno, Laura Farinetti, and Alessio Bosca. Ontology driven semantic search. *WSEAS Transaction on Information Science and Application*, 1(6):1597–1605, 2004.

[BCM+03] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003.

[BCS+07] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open information extraction from the web. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2670–2676, 2007.

[BCT07] Karin Breitman, Marco Antonio Casanova, and Walt Truszkowski. *Semantic Web: Concepts, Technologies and Applications.* Springer-Verlag Berlin Heidelberg New York, 2007.

[BDF+06] Piero A. Bonatti, Claudiu Duma, Norbert Fuchs, Wolfgang Nejdl, Daniel Olmedilla, Joachim Peer, and Nahid Shahmehri. Semantic web policies - a discussion of requirements and research issues. In *European Semantic Web Conference (ESWC)*, pages 712–724, 2006.

[BH90] David M. Bridgeland and Michael N. Huhns. Distributed truth maintenance. In *American Association for Artificial Intelligence (AAAI)*, pages 72–77, 1990.

[BH06] Alexander Budanitsky and Graeme Hirst. Evaluating Wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

[BHyP+06] Boualem Benatallah, Mohand-Said Hacid, Hye young Paik, Christophe Rey, and Farouk Toumani. Towards semantic-driven, flexible and scalable framework for peering and querying e-catalog communities. *Information Systems*, 31(4):266–294, 2006.

[BKM99] Franz Baader, Ralf Küsters, and Ralf Molitor. Computing least common subsumers in description logics with existential restrictions. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 96–101, 1999.

[BKM00] Franz Baader, Ralph Küsters, and Ralf Molitor. Rewriting concepts using terminologies. In *Knowledge Representation (KR)*, pages 297–308, 2000.

[BKT02] Sebastian Brandt, Ralf Küsters, and Anni-Yasmin Turhan. Approximation and difference in description logics. In *Knowledge Representation (KR)*, pages 203–214, 2002.

[BLHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):35–43, 2001.

[Blo01] François-Marie Blondel. La recherche d'informations sur internet par des lycéens, analyse et assistance à l'apprentissage. In *Hypermédias et Apprentissages*, pages 119–133, 2001.

[Bon01] Curtis Bonk. *Online Teaching in an Online World*. CourseShare.com, `http://www.publicationshare.com/`, 2001.

[Bra06] Sebastian Brandt. *Standard and Non-Standard Reasoning in Description Logics*. PhD thesis, Technische Universität Dresden, 2006.

[Bro02] Andrei Z. Broder. A taxonomy of web search. *ACM SIGIR Forum*, 36(2):3–10, 2002.

[BSZ04] Paolo Bouquet, Luciano Serafini, and Stefano Zanobini. Peer-to-peer semantic coordination. *Web Semantics*, 2(1):81–97, 2004.

[Bur98] Pamela Burdman. Classrooms without walls: More students are taking college courses online. *San Francisco Chronicle*, July(20):A–1, 1998.

[BVL03] Sean Bechhofer, Raphael Volz, and Phillip W. Lord. Cooking the semantic web with the OWL API. In *International Semantic Web Conference (ISWC)*, pages 659–675, 2003.

[BWH05] Alexander Borgida, Thomas Walsh, and Haym Hirsh. Towards measuring similarity in description logics. In *Description Logics (DL)*, 2005.

[BYRN99] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval.* ACM Press / Addison-Wesley, 1999.

[CDD+04] Jeremy J. Carroll, Ian Dickinson, Chris Dollin, Dave Reynolds, Andy Seaborne, and Kevin Wilkinson. Jena: implementing the semantic web recommendations. In *WWW - Alternate Track Papers & Posters*, pages 74–83, 2004.

[CE05] Michael J. Cafarella and Oren Etzioni. A search engine for natural language applications. In *WWW*, pages 442–452, 2005.

[CEE+01] Kai-Uwe Carstenense, Christian Ebert, Cornelia Endriss, Susanne Jekat, Ralf Klabunde, and Hagen Langer, editors. *Computerlinguistik und Sprachentechnologie: Eine Einführung.* Spektrum Akademsicher Verlag, Heidelberg-Berlin, 2001.

[CGPL+03] Óscar Corcho, Asunción Gómez-Pérez, Alain Leger, Christophe Rey, and Farouk Toumani. An ontology-based mediation architecture for e-commerce applications. In *Intelligent Information Processing and Web Mining (IIPWM)*, pages 477–486, 2003.

[CH03] Yu Chen and Wei Jyh Heng. Automatic synchronization of speech transcript and slides in presentation. In *International Symposium on Circuits and Systems (ISCAS)*, pages 568–571, 2003.

[CMZ03] Mark G. Core, Johanna D. Moore, and Claus Zinn. The role of initiative in tutorial dialogue. In *European Chapter of the Association for Computational Linguistics (EACL)*, pages 67–74, 2003.

[CNS+03] Simona Colucci, Tommaso Di Noia, Eugenio Di Sciascio, Francesco M. Donini, and Marina Mongiello. Concept abduction and contraction in description logics. In *International Workshop on Description Logics (DL)*, 2003.

[CNS+05a] Simona Colucci, Tommaso Di Noia, Eugenio Di Sciascio, Francesco M. Donini, and Azzurra Ragone. Semantic-based automated composition of distributed learning objects for personalized e-learning. In *European Semantic Web Conference (ESWC)*, pages 633–648, 2005.

[CNS+05b] Simona Colucci, Tommaso Di Noia, Eugenio Di Sciascio, Francesco M. Donini, and Azzurra Ragone. *The Semantic Web: Research and Applications*, chapter Semantic-Based Automated Composition of Distributed Learning Objects for Personalized E-Learning, pages 633–648. Springer-Verlag Berlin Heidelberg New York, 2005.

[CNS+05c] Stefano Coppi, Tommaso Di Noia, Eugenio Di Sciascio, Francesco M. Donini, and Agnese Pinto. Ontology-based natural language parser for e-marketplaces. In *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE)*, pages 279–289, 2005.

[CRFRJFN05] Jinwei Cao, Jose Antonio Robles-Flores, Dmitri Roussinov, and Jr. Jay F. Nunamaker. Automated question answering from lecture videos: NLP vs. pattern matching. In *Hawaii International Conference on System Sciences (HICSS)*, 2005.

[DBB+02] Susan Dumais, Michele Banko, Eric Brill, Jimmy Lin, and Andrew Ng. Web question answering: is more always better? In *ACM SIGIR*, pages 291–298, 2002.

[dFE06] Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. A dissimilarity measure for ALC concept descriptions. In *ACM SIGSAC*, pages 1695–1699, 2006.

[Dit03] Ulrich Dittler, editor. *E-Learning - Einsatzkonzepte und Erfolgsfaktoren des Lernens mit interaktiven Medien.* Oldenbourg Wissenschaftsverlag, München, 2003.

[dK86] Johan de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28(2):127–162, 1986.

[Dow05] Stephen Downes. E-learning 2.0. *e-Learn Magazine*, 2005.

[ECD+04] Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Web-scale information extraction in knowitall. In *WWW*, pages 100–110, 2004.

[Eer06] Matthew Eernisse. *Build Your Own Ajax Web Applications.* Sitepoint, 2006.

[EHLS06] Michael Engelhardt, Arne Hildebrand, Dagmar Lange, and Thomas C. Schmidt. Reasoning about elearning multimedia objects. In *Semantic Web Annotations for Multimedia (SWAMM)*, 2006.

[ES07] Jérôme Euzenat and Pavel Shvaiko. *Ontology matching.* Springer-Verlag Berlin Heidelberg New York, 2007.

[FDD+99] Raya Fidel, Rachel K. Davies, Mary H. Douglass, Jenny K. Holder, Carla J. Hopkins, Elisabeth J. Kushner, Bryan K. Miyagishima, and Christina D. Toney. A visit to the information mall: Web searching behavior of high school students. *American Society for Information Science*, 50(1):24–37, 1999.

[Fen04] Dieter Fensel. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce.* Springer-Verlag Berlin Heidelberg New York, 2004.

[Fra96] Enrico Franconi. Logical form and knowledge representation: towards a reconciliation. In *Knowledge Representation Systems based on Natural Language*, pages 20–24, 1996.

[Fra03] Enrico Franconi. *The Description Logic Handbook: Theory, Implementation, and Applications*, chapter Natural Language Processing, pages 450–461. Cambridge University Press, 2003.

[FSS03] Wellesley Foshay, Kenneth Silber, and Michael Stelnicki. *Writing Training Materials That Work: How to Train Anyone to Do Anything.* Pfeiffer, 2003.

[GH07] Hannah Green and Celia Hannon. *Their Space: Education for a Digital Generation.* Demos, 2007.

[GMP06] Stephan Grimm, Boris Motik, and Chris Preist. Matching semantic service descriptions with local closed-world reasoning. In *European Semantic Web Conference (ESWC)*, pages 575–589, 2006.

[GPCGFL03] Asuncion Gomez-Perez, Oscar Corcho-Garcia, and Mariano Fernandez-Lopez. *Ontological Engineering.* Springer-Verlag Berlin Heidelberg New York, 2003.

[Gru93] Thomas R. Gruber. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199–220, 1993.

[Gru95] Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing? *Human-Computer Studies*, 43(5-6):907–928, 1995.

[Har96] Andre Hardy. On the number of clusters. *Computational Statistics and Data Analysis*, 23:83–96, 1996.

[HG01] Lynette Hirschman and Rob Gaizauskas. Natural language question answering: The view from here. *Natural Language Engineering*, 7(4):275–300, 2001.

[Hje01] Johan Hjelm. *Creating the Semantic Web with RDF: Professional Developer's Guide.* John Wiley & Sons, 2001.

[HK05] Alexander Haubold and John R. Kender. Augmented segmentation and visualization for presentation videos. In *ACM SIGMM*, pages 51–60, 2005.

[HKW02] Wolfgang Hürst, Thorsten Kreuzer, and Marc Wiesenhütter. A qualitative study towards using large vocabulary automatic speech recognition to index recorded presentations for search and access over the web. In *IADIS Internatinal Conference WWW/Internet (ICWI)*, pages 135–143, 2002.

[HLRT02] Mohand-Saïd Hacid, Alain Leger, Christophe Rey, and Farouk Toumani. Computing concept covers: A preliminary report. In *Description Logics (DL)*, 2002.

[HM01] Volker Haarslev and Ralf Möller. Racer system description. In *International Joint Conference on Automated Reasoning (IJCAR)*, pages 701–706, 2001.

[HM02] Elliotte Rusty Harold and W. Scott Means. *XML in a Nutshell, 2nd Edition.* O'Reilly, 2002.

[HMS05] Denis Helic, Hermann Maurer, and Nick Scerbakov. A didactics aware approach to knowledge transfer in web-based education. *Studies in Fuzziness and Soft Computing*, 178:233–260, 2005.

[Hor98] Ian Horrocks. The FaCT system. In *Automated Reasoning with Analytic Tableaux and Related Methods: International Conference Tableaux*, pages 307–312, 1998.

[HPM05] Andreas Holzinger, Arnold Pichler, and Hermann Maurer. Multi media e-learning software triangle case-study: Experimental results and lessons learned. *Universal Science and Technology of Learning*, 0(0):61–92, 2005.

[HPS04] Ian Horrocks and Peter Patel-Schneider. Reducing OWL entailment to description logic satisfiability. *Journal of Web Semantics*, 1(4):345–357, 2004.

[HS00] Christoph Hölscher and Gerhard Strube. Web search behavior of internet experts and newbies. *Computer Networks*, 33(1-6):337–346, 2000.

[HV05] Peter Haase and Johanna Völker. Ontology learning and reasoning - dealing with uncertainty and inconsistency. In *Uncertainty Reasoning for the Semantic Web (URSW)*, pages 45–55, 2005.

[Inm04] Dave Inman. *The possibility of Natural Language Processing by computer.* `http://www.scism.sbu.ac.uk/inmandw/tutorials/nlp/intro/intro.html`, 2004.

[JNL+05] Alejandro Jaimes, Takeshi Nagamine, Jianyi Liu, Kengo Omura, and Nicu Sebe. Affective meeting video analysis. In *Multimedia and Expo*, pages 1412–1415, 2005.

[Kat97] Boris Katz. Annotating the world wide web using natural language. In *Computer Assisted Information Searching on the Internet (RIO)*, 1997.

[KBHS04] Naouel Karam, Salima Benbernou, Mohand-Said Hacid, and Michel Schneider. Semantic matching of natural language web queries. In *International Conference on Web Engineering (ICWE)*, pages 416–429, 2004.

[KBZ06] Esther Kaufmann, Abraham Bernstein, and Renato Zumstein. Querix: A natural language interface to query ontologies based on clarification dialogs. In *International Semantic Web Conference (ISWC)*, pages 980–981, 2006.

[KEW01] Cody C. T. Kwok, Oren Etzioni, and Daniel S. Weld. Scaling question answering to the web. In *WWW*, pages 150–161, 2001.

[KFNM04] Holger Knublauch, Ray W. Fergerson, Natalya Fridman Noy, and Mark A. Musen. The protégé OWL plugin: An open development environment for semantic web applications. In *International Semantic Web Conference (ISWC)*, pages 229–243, 2004.

[KFY+02] Boris Katz, Sue Felshin, Deniz Yuret, Ali Ibrahim, Jimmy J. Lin, Gregory Marton, Alton Jerome McFarland, and Baris Temelkuran. Omnibase: Uniform access to heterogeneous data for question answering. In *Applications of Natural Language to Information Systems (NLDB)*, pages 230–234, 2002.

[KL02] Boris Katz and Jimmy Lin. Annotating the semantic web using natural language. In *NLP and XML (NLPXML)*, 2002.

[KLM07] Naouel Karam, Serge Linckels, and Christoph Meinel. Semantic composition of lecture subparts for a personalized e-learning. In *European Semantic Web Conference (ESWC)*, pages 716–728, 2007.

[KMR04] Holger Knublauch, Mark A. Musen, and Alan L. Rector. Editing description logic ontologies with the protégé OWL plugin. In *Description Logics (DL)*, 2004.

[KPT+04] Atanas Kiryakov, Borislav Popov, Ivan Terziev, Dimitar Manov, and Damyan Ognyanoff. Semantic annotation, indexing, and retrieval. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2(1):49–79, 2004.

[KS05] Dimitris J. Kavvadias and Elias C. Stavropoulos. An efficient algorithm for the transversal hypergraph generation. *Graph Algorithms Applications*, 9(2):239–264, 2005.

[Kup93] Julian Kupiec. Murax: A robust linguistic approach for question answering using an on-line encyclopedia. In *ACM SIGIR*, pages 181–190, 1993.

[Küs01] Ralf Küsters. *Non-Standard Inferences in Description Logics*. Springer-Verlag Berlin Heidelberg New York, 2001.

[Lac05] Lee W. Lacy. *OWL: Representing Information Using the Web Ontology Language*. Trafford Publishing, 2005.

[LC02] Xiaoyong Liu and W. Bruce Croft. Passage retrieval based on language models. In *Conference on Information and Knowledge Management (CIKM)*, pages 375–382, 2002.

[LDM06] Serge Linckels, Carole Dording, and Christoph Meinel. Better results in mathematics lessons with a virtual personal teacher. In *ACM SIGUCCS*, pages 201–209, 2006.

[LDM07] Serge Linckels, Carole Dording, and Christoph Meinel. Bessere Schulnoten mit MatES, dem e-Bibliothekardienst für den Mathematikunterricht. In *Deutsche e-Learning Fachtagung Informatik der Gesellschaft für Informatik (DeLFI)*, pages 91–102, 2007.

[Lew05] Dirk Lewandowski. *Informationswissenschaft*. Deutsche Gesellschaft für Informationswissenschaft und Informationspraxis e.V., 2005.

[LH99] Tessa Lau and Eric Horvitz. Patterns of search: Analyzing and modeling web query refinement. In *User Modeling*, 1999.

[LKM06] Serge Linckels, Mathias Kutzner, and Christoph Meinel. tele-TASK und CHESt - zwei moderne werkzeuge im alltäglichen unterricht. *Online Tutoring Journal*, 1(1), 2006.

[LM04a] Serge Linckels and Christoph Meinel. An application of semantics for an educational tool. In *IADIS Applied Computing (AC)*, pages II:234–II:239, 2004.

[LM04b] Serge Linckels and Christoph Meinel. Automatic interpretation of natural language for a multimedia e-learning tool. In *International Conference on Web Engineering (ICWE)*, pages 435–439, 2004.

[LM04c] Serge Linckels and Christoph Meinel. An educational tool that understands students' questions. In *Association for Educational Communications and Technology (AECT)*, pages 435–444, 2004.

[LM04d] Serge Linckels and Christoph Meinel. Semantic layer architecture for an educational expert system in computer history. In *Advances in Intelligent Systems Theory and Applications (AISTA)*, 2004.

[LM05a] Serge Linckels and Christoph Meinel. An educational tool that understands students' questions. *Distance Learning*, 2(3):13–21, 2005.

[LM05b] Serge Linckels and Christoph Meinel. A simple application of description logics for a semantic search engine. In *IADIS Applied Computing (AC)*, volume 2, pages 306–311, 2005.

[LM05c] Serge Linckels and Christoph Meinel. A simple solution for an intelligent librarian system. In *IADIS Applied Computing (AC)*, volume 1, pages 495–503, 2005.

[LM06a] Serge Linckels and Christoph Meinel. An e-librarian service - natural language interface for an efficient semantic search within multimedia resources. Technischer Bericht 12, Hasso-Plattner-Institut (HPI) für Softwaresystemtechnik GmbH, 2006.

[LM06b] Serge Linckels and Christoph Meinel. An e-librarian service that yields pertinent resources from a multimedia knowledge base. In *Signal Processing and Multimedia Applications (SIGMAP)*, pages 208–215, 2006.

[LM06c] Serge Linckels and Christoph Meinel. Resolving ambiguities in the semantic interpretation of natural language questions. In *Intelligent Data Engineering and Automated Learning (IDEAL)*, pages 612–619, 2006.

[LM07] Serge Linckels and Christoph Meinel. Semantic interpretation of natural language user input to improve search in multimedia knowledge base. *it - Information Technology*, 1(49):40–48, 2007.

[LME05] Serge Linckels, Christoph Meinel, and Thomas Engel. Teaching in the cyber-age: Technologies, experiments, and realizations. In *Deutsche e-Learning Fachtagung Informatik der Gesellschaft für Informatik (DeLFI)*, pages 225–236, 2005.

[LME06] Serge Linckels, Christoph Meinel, and Thomas Engel. An e-librarian service that understands natural language questions from students. In *Conference on Advanced Information Systems Engineering (CAiSE) - Forum paper*, 2006.

[LPM05] Vanessa Lopez, Michele Pasin, and Enrico Motta. Aqualog: An ontology-portable question answering system for the semantic web. In *European Semantic Web Conference (ESWC)*, pages 546–562, 2005.

[LRKM07] Serge Linckels, Stephan Repp, Naouel Karam, and Christoph Meinel. The virtual tele-task professor: semantic search in recorded lectures. In *ACM SIGCSE*, pages 50–54, 2007.

[LSM07a] Serge Linckels, Harald Sack, and Christoph Meinel. Optimizing the retrieval of pertinent answers for nl questions with the e-librarian service. In *Service Matchmaking and Resource Retrieval in the Semantic Web (SMR2)*, 2007.

[LSM07b] Serge Linckels, Harald Sack, and Christoph Meinel. Retrieval optimization of pertinent answers for nl questions with the e-librarian service. In *International Semantic Web Conference (ISWC) poster and demo track*, 2007.

[LTV03] Ulrike Lucke, Djamshid Tavangarian, and Denny Voigt. Multidimensional educational multimedia with <ml>^{3}. In *E-Learning in Corporate, Government, Healthcare, and Higher Education (ELEARN)*, 2003.

[Luc06] Ulrike Lucke. An algebra for multidimensional documents as abstraction mechanism for cross media publishing. In *Automated Production of Cross Media Content for Multi-Channel Distribution*, pages 165–172, 2006.

[Ma98] George A. Miller and al. *WordNet: An Electronic Lexical Database.* The MIT Press, 1998.

[Mar03] Philippe Martin. *Web Intelligence*, chapter Knowledge Representation, Sharing and Retrieval on the Web, pages 263–297. Springer-Verlag Berlin Heidelberg New York, 2003.

[MBR01] Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. Generic schema matching with Cupid. In *Very Large Databases (VLDB)*, pages 49–58, 2001.

[MG90] Richard E. Mayer and Joan K. Gallini. When is an illustration worth ten thousand words? *Educational Psychology*, 82(4):715–726, 1990.

[MHG+02] Dan I. Moldovan, Sanda M. Harabagiu, Roxana Girju, Paul Morarescu, V. Finley Lacatusu, Adrian Novischi, Adriana Badulescu, and Orest Bolohan. LCC tools for question answering. In *Text Retrieval Conference (TREC)*, 2002.

[Mit04] Ruslan Mitkov, editor. *The Oxford Handbook of Computational Linguistics*. Oxford University Press, 2004.

[MJ89] Cindy L. Mason and Rowland R. Johnson. *Distributed Artificial Intelligence*, volume 2, chapter DATMS: a framework for distributed assumption based reasoning, pages 293–317. Morgan Kaufmann Publishers Inc., San Francisco, 1989.

[Mor05] Peter Morville. *Ambient Findability - What We Find Changes Who We Become*. O'Reilly Media, Inc., 2005.

[MS94] Richard E. Mayer and Valerie K. Sims. For whom is a picture worth a thousand words? Extensions of a dual-coding theory of multimedia learning. *Educational Psychology*, 86(3):389–401, 1994.

[MS99] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

[MS04a] Christoph Meinel and Harald Sack. *WWW - Kommunikation, Internetworking, Web-Technologien*. Springer-Verlag Berlin Heidelberg New York, 2004.

[MS04b] Erica Melis and Jörg H. Siekmann. Activemath: An intelligent tutoring system for mathematics. In *Artificial Intelligence and Soft Computing (ICAISC)*, pages 91–101, 2004.

[MSMV04] Robert Mertens, Holger Schneider, Olaf Müller, and Oliver Vornberger. Hypermedia navigation concepts for lecture recordings. In *E-Learning in Corporate, Government, Healthcare, and Higher Education (ELEARN)*, pages 2480–2847, 2004.

[MSR02] Libby Miller, Andy Seaborne, and Alberto Reggiori. Three implementations of SquishQL, a simple RDF query language. In *International Semantic Web Conference (ISWC)*, pages 423–435, 2002.

[MSS05] Boris Motik, Ulrike Sattler, and Rudi Studer. Query answering for OWL-DL with rules. *Semantic Web*, 3(1):41–60, 2005.

[MZ07] Yuval Marom and Ingrid Zukerman. A predictive approach to help-desk response generation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1665–1670, 2007.

[Neb90] Bernhard Nebel. *Reasoning and revision in hybrid representation systems.* Springer-Verlag Berlin Heidelberg New York, 1990.

[NPSR99] Raquel Navarro-Prieto, Mike Scaife, and Yvonne Rogers. Cognitive strategies in web searching. In *Conference on Human Factors & the Web*, 1999.

[NSDM03] Tommaso Di Noia, Eugenio Di Sciascio, Francesco M. Donini, and Marina Mongiello. Abductive matchmaking using description logics. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 337–342, 2003.

[NWP03] Chong-Wah Ngo, Feng Wang, and Ting-Chuen Pong. Structuring lecture videos for distance learning applications. In *Multimedia Software Engineering*, pages 215–222, 2003.

[OAKS04] Kieron O'Hara, Harith Alani, Yannis Kalfoglou, and Nigel Shadbolt. Trust strategies for the semantic web. In *Trust, Security, and Reputation on the Semantic Web*, 2004.

[ÖS05] Annika Öhgren and Kurt Sandkuhl. Towards a methodology for ontology development in small and medium-sized enterprises. In *IADIS Applied Computing (AC)*, pages 369–376, 2005.

[Ows97] Ronald D. Owston. The world wide web: A technology to enhance teaching and learning? *Educational Researcher*, 26(2):27–33, 1997.

[PBR06] Jane C. Prey, Dave Berque, and Robert H. Reed. *The Impact of Tablet PCs and Pen-based Technology on Education: Vignettes, Evaluations, and Future Directions.* Purdue University Press, 2006.

[PEK03] Ana-Maria Popescu, Oren Etzioni, and Henry A. Kautz. Towards a theory of natural language interfaces to databases. In *Intelligent User Interfaces (IUI)*, pages 149–157, 2003.

[PM05] Luiz Augusto Pizzato and Diego Molla. Extracting exact answers using a meta question answering system. In *Australasian Language Technology Workshop (ALTA)*, 2005.

[Pop05] Octav Popescu. *Logic-Based Natural Language Understanding in Intelligent Tutoring Systems.* PhD thesis, Carnegie Mellon University, 2005.

[Pow03] Shelley Powers. *Practical RDF. Solving Problems with the Resource Description Framework.* O'Reilly Media, 2003.

[QYJ04] Yang Qing, Yang Yang, and Chen Juan. Goal-oriented platform based on knowledge-point: A new model of distance education system. In *Advanced Information Networking and Applications (AINA)*, pages 528–531, 2004.

[RCPB04] Ganesh Ramakrishnan, Soumen Chakrabarti, Deepa Paranjpe, and Pushpak Bhattacharya. Is question answering an acquired skill? In *WWW*, pages 111–120, 2004.

[RDH+04] Alan L. Rector, Nick Drummond, Matthew Horridge, Jeremy Rogers, Holger Knublauch, Robert Stevens, Hai Wang, and Chris Wroe. OWL pizzas: Practical experience of teaching OWL-DL: Common errors & common patterns. In *Engineering Knowledge in the Age of the Semantic Web (EKAW)*, pages 63–81, 2004.

[Res05] Sherri L. Restauri. Creating an effective online distance education program using targeted support factors. *AECT TechTrends*, 48(6):32–39, 2005.

[RLM05] Monique Reichert, Serge Linckels, and Christoph Meinel. Student's perception of a semantic search engine. In *Cognition and Exploratory Learning in Digital Age (CELDA)*, pages 139–147, 2005.

[RLM07] Stephan Repp, Serge Linckels, and Christoph Meinel. Towards to an automatic semantic annotation for multimedia learning objects. In *Educational Multimedia and Multimedia Education (EMME)*, 2007.

[RM06] Stephan Repp and Christoph Meinel. Semantic indexing for recorded educational lecture videos. In *International Conference on Pervasive Computing and Communications Workshops (PERCOMW)*, page 240, 2006.

[RRF04] Dmitri Roussinov and Jose Antonio Robles-Flores. Learning patterns to answer open domain questions on the web. In *ACM SIGIR*, pages 500–501, 2004.

[RRFD04] Dmitri Roussinov, José Antonio Robles-Flores, and Yin Ding. Experiments with web QA system and TREC 2004 questions. In *Text Retrieval Conference (TREC)*, 2004.

[Sch93] Renate A. Schmidt. Terminological representation, natural language & relation algebra. In *German AI Conference (GWAI)*, pages 357–371, 1993.

[Sch03] Sabine Schulte im Walde. *Experiments on the Automatic Induction of German Semantic Verb Classes.* PhD thesis, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart, 2003.

[SF99] Christian Süß and Burkhard Freitag. Metamodeling for web-based teachware management. In *Advances in Conceptual Modeling*, pages 360–373, 1999.

[SM02] Volker Schillings and Christoph Meinel. tele-TASK: teleteaching anywhere solution kit. In *ACM SIGUCCS*, pages 130–133, 2002.

[SMMS02] Ljiljana Stojanovic, Alexander Maedche, Boris Motik, and Nenad Stojanovic. User-driven ontology evolution management. In *Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web (EKAW)*, pages 285–300, 2002.

[SP04] Evren Sirin and Bijan Parsia. Pellet: An OWL DL reasoner. In *Description Logics (DL)*, 2004.

[SS04] Steffen Staab and Rudi Studer, editors. *Handbook on Ontologies.* International Handbooks on Information Systems. Springer-Verlag Berlin Heidelberg New York, 2004.

[SSS91] Manfred Schmidt-Schauß and Gert Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[SW06a] Harald Sack and Jörg Waitelonis. Automated annotations of synchronized multimedia presentations. In *Mastering the Gap: From Information Extraction to Semantic Representation*, 2006.

[SW06b] Harald Sack and Jörg Waitelonis. Integrating social tagging and document annotation for content-based search in multimedia data. In *Semantic Authoring and Annotation Workshop (SAAW)*, 2006.

[SW06c] Harald Sack and Jörg Waitelonis. Integrating social tagging and document annotation for content-based search in multimedia data. In *Semantic Authoring and Annotation Workshop (SAAW)*, 2006.

[Tee94] Gunnar Teege. Making the difference: a subtraction operation for description logics. In *Knowledge Representation (KR)*, pages 540–550, 1994.

[TFG05] Allison Elliott Tew, Charles Fowler, and Mark Guzdial. Tracking an innovation in introductory CS education from a research university to a two-year college. In *ACM SIGCSE*, pages 416–420, 2005.

[TGF+07] Javier Tejedor, Roberto García, Miriam Fernández, Fernando J. López-Colino, Ferrán Perdrix, José A. Macías, Rosa M. Gil, Marta Oliva, Diego Moya, José Colás, and Pablo Castells. Ontology-based retrieval of human speech. In *Web Semantics (WebS)*, 2007.

[TM01] Lappoon R. Tang and Raymond J. Mooney. Using multiple clause constructors in inductive logic programming for semantic parsing. In *European Conference on Machine Learning (ECML)*, pages 466–477, 2001.

[Tro03] Raphaël Troncy. Integrating structure and semantics into audio-visual documents. In *International Semantic Web Conference (ISWC)*, pages 566–581, 2003.

[VdSA06] Daniel Schwabe Vinicius da Silva Almendra. Trust policies for semantic web repositories. In *Semantic Web Policy Workshop (SWPW)*, 2006.

[VVSH07] Johanna Völker, Denny Vrandecic, York Sure, and Andreas Hotho. Learning disjointness. In *European Semantic Web Conference (ESWC)*, pages 175–189, 2007.

[WAC84] Robert Wilensky, Yigal Arens, and David Chin. Talking to unix in english: an overview of UC. *Communications of the ACM*, 27(6):574–593, 1984.

[War06a] Mark Warschauer. Going one-to-one. *Educational Leadership*, 63(4):34–38, 2006.

[War06b] Mark Warschauer. *Laptops and Literacy: Learning in the Wireless Classroom.* Teachers College, 2006.

[WBFT04] Dapeng Wang, Thomas Bayer, Thilo Frotscher, and Marc Teufel. *Java Web Services mit Apache Axis.* Software & Support Verlag, Frankfurt, 2004.

[WFSP00] Leejay Wu, Christos Faloutsos, Katia P. Sycara, and Terry R. Payne. FALCON: Feedback adaptive loop for content-based retrieval. In *Verly Large Databases (VLDB)*, pages 297–306, 2000.

[WLM07] Katrin Wolf, Serge Linckels, and Christoph Meinel. Teleteaching anywhere solution kit (tele-task) goes mobile. In *ACM SIGUCCS*, 2007.

[WLZR07] René Witte, Qiangqiang Li, Yonggang Zhang, and Juergen Rilling. Ontological text mining of software documents. In *Applications of Natural Language to Information Systems (NLDB)*, pages 168–180, 2007.

[WMSB01] Wendy M. Williams, Faith Markle, Robert J. Sternberg, and Melanie Brigockas. *Educational Psychology.* Allyn & Bacon, 2001.

[WNP06] Feng Wang, Chong-Wah Ngo, and Ting-Chuen Pong. Prediction-based gesture detection in lecture videos by combining visual, speech and electronic slides. In *Multimedia and Expo*, pages 653–656, 2006.

[WPS+04] Patrick Wolf, Wolfgang Putz, Avare Stewart, Arnd Steinmetz, Matthias Hemmje, and Erich Neuhold. Lecturelounge – experience education beyond the borders of the classroom. *Digital Libraries*, 4(1):39–41, 2004.

[YH03] Christine Youngblut and Odette Huie. The relationship between presence and performance in virtual environments: Results of a verts study. In *Virtual Reality (VR)*, 2003.

[YOA03] Natsuo Yamamoto, Jun Ogata, and Yasuo Ariki. Topic segmentation and retrieval system for lecture videos based on spontaneous speech recognition. In *European Conference on Speech Communication and Technology*, pages 961–964, 2003.

[You98] Christine Youngblut. Educational uses of virtual reality technology. Technical Report IDA Document D-2128, Defense Advanced Research Projects Agency, `http://www.hitl.washington.edu/scivw/youngblut-edvr/D2128.pdf`, 1998.

[Zhe02] Zhiping Zheng. Developing a web-based question answering system. In *WWW - poster*, 2002.

[ZZ03] Yingying Zhu and Dongru Zhou. Video browsing and retrieval based on multimodal integration. In *Web Intelligence*, pages 650–653, 2003.

Part V

Appendix

Appendix A

References

ACM	Association for Computing Machinery
AECT	Association for Educational Communications and Technology
AI	Artificial Intelligence
ALIWEB	Archie Like Index for the Web
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BC	Before Christ
CD	Compact Disc
CFG	Context Free Grammar
CHESt	Computer History Expert System
CPU	Central Processing Unit
CWA	Closed World Assumption
DAML	DARPA Agent Markup Language
DARPA	Defense Advanced Research Projects Agency
DBMS	Database Management System
DL	Description Logics
DLP	Description Logic Programs
DVD	Digital Video Disc
FTP	File Transfer Protocol
GFO	General Formal Ontology
GHz	Gigahertz
GUI	Graphical User Interface
HL-PCFG	Head-Lexicalized Probabilistic Context-Free Grammar
HCI	Human Computer Interface
HPI	Hasso-Plattner-Institut
HTTP	Hypertext Transfer Protocol
HTTPS	Secure HTTP
ID	Identifier
IR	Information Retrieval
IRI	Internationalized Resource Identifier
ISBN	International Standard Book Number

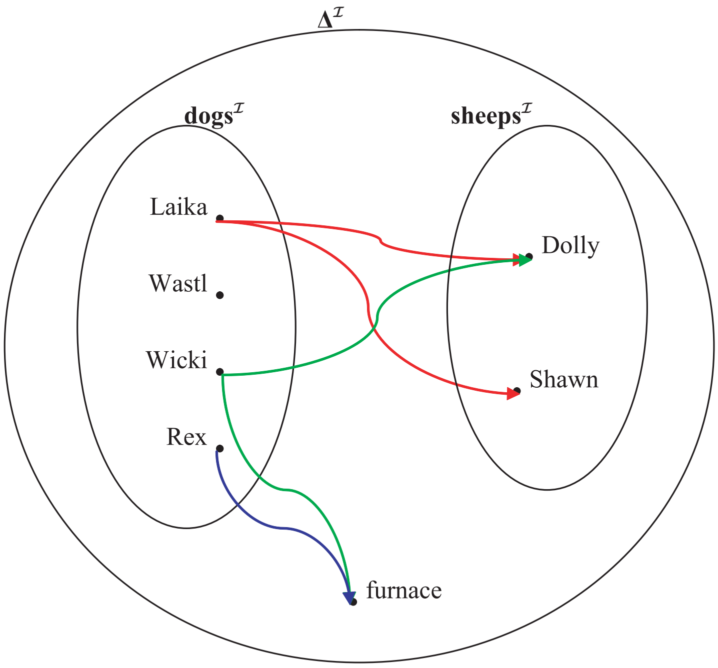
KIF	Knowledge Interchange Format
LAN	Local Area Network
LCS	Least Common Subsumer
LO	Learning Object
LTE	Lycée Technique Esch/Alzette
NL	Natural Language
NLP	Natural Language Processing
MatES	Mathematics Expert System
MB	Megabyte
MERLOT	Multimedia Educational Resource for Learning and Online Teaching
MOM	Multimedia Ontology Manager
MSC	Most Specific Concept
MIR	Multimedia Information Retrieval
MIT	Massachusetts Institute of Technology
OIL	Ontology Inference Layer or Ontology Interchange Language
OWA	Open World Assumption
OWL	Ontology Web Language
PC	Personal Computer
PDA	Personal Digital Assistant
Ph.D.	Doctor of Philosophy (Latin: *Philosophiae Doctor*)
POS tagger	Part-Of-Speech tagger
QA	Question-Answering
RAM	Random Access Memory
RDF	Resource Description Framework
RDFS	RDF Schema
RDQL	RDF Query Language
RIF	Rule Interchange Format
ROM	Read Only Memory
RPC	Remote Procedure Call
RSS	Really Simple Syndication
SGML	Standard Generalized Markup Language
SMTP	Simple Mail Transfer Protocol
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol or Service Oriented Architecture Protocol
SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language
SUMO	Suggested Upper Merged Ontology
SW	Semantic Web
SWRL	Semantic Web Rule Language
TCP/IP	Transmission Control Protocol / Internet Protocol
tele-TASK	Teleteaching Anywhere Solution Kit
TREC	Text Retrieval Conference
UC	Unix Consultant

URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WKW	Well-Known-Words
WLH	World Lecture Hall
WSDL	Web Services Description Language
WWW	World Wide Web
XML	Extensible Markup Language
XMLS	XML Schema
XSLT	Extensible Stylesheet Language Transformations

Appendix B

Example of an Interpretation in Description Logics

$$\begin{array}{rcl}
\Delta^{\mathcal{I}} & = & \{\text{Laika,Wastl,Wicki,Rex,Dolly,Shawn,furnace}\} \\
Dog^{\mathcal{I}} & = & \{Laika, Wastl, Wicki, Rex\} \\
Sheep^{\mathcal{I}} & = & \{Dolly, Shawn\} \\
guard^{\mathcal{I}} & = & \{(Laika, Dolly), (Laika, Shawn), (Wicki, Dolly), (Wicki, furnace), (Rex, furnace)\}
\end{array}$$



First example ($\mathcal{FL}^-$): A sheepdog is a dog that among other things, guards something.

$$\begin{array}{rcl}
Sheepdog & \equiv & Dog \sqcap \exists guard.\top \\
Sheepdog^{\mathcal{I}} & = & \{a \in Dog^{\mathcal{I}}\} \cap \{a \in \Delta^{\mathcal{I}} | \exists b.(a,b) \in guard^{\mathcal{I}}\} \\
 & = & \{Laika, Wicki, Rex\}
\end{array}$$

Second example ($\mathcal{FL}_0$): A sheepdog is a dog that among other things, guards only sheep.

$$\begin{aligned} Sheepdog &\equiv Dog \sqcap \forall guard.Sheep \\ Sheepdog^{\mathcal{I}} &= \{a \in Dog^{\mathcal{I}}\} \cap \{a \in \Delta^{\mathcal{I}} | \forall b.(a,b) \in guard^{\mathcal{I}} \rightarrow b \in Sheep^{\mathcal{I}}\} \\ &= \{Laika, Wastl\} \end{aligned}$$

Third example ($\mathcal{FL}_0$): A sheepdog is a dog that among other things, guards nothing.

$$\begin{aligned} Sheepdog &\equiv Dog \sqcap \forall guard.\bot \\ Sheepdog^{\mathcal{I}} &= \{a \in Dog^{\mathcal{I}}\} \cap \{a \in \Delta^{\mathcal{I}} | \forall b.(a,b) \in guard^{\mathcal{I}} \rightarrow b \in \emptyset\} \\ &= \{Wastl\} \end{aligned}$$

Fourth example ($\mathcal{ALE}$): A sheepdog is a dog that among other things, guards sheep.

$$\begin{aligned} Sheepdog &\equiv Dog \sqcap \exists guard.Sheep \\ Sheepdog^{\mathcal{I}} &= \{a \in Dog^{\mathcal{I}}\} \cap \{a \in \Delta^{\mathcal{I}} | \exists b.(a,b) \in guard^{\mathcal{I}} \wedge b \in Sheep^{\mathcal{I}}\} \\ &= \{Laika, Wicki\} \end{aligned}$$

Fifth example ($\mathcal{ALE}$): A sheepdog is a dog that among other things, guards at least nothing.

$$\begin{aligned} Sheepdog &\equiv Dog \sqcap \exists guard.\bot \\ Sheepdog^{\mathcal{I}} &= \{a \in Dog^{\mathcal{I}}\} \cap \{a \in \Delta^{\mathcal{I}} | \exists b.(a,b) \in guard^{\mathcal{I}} \wedge b \in \bot^{\mathcal{I}}\} \\ &= \text{always inconsistent} \end{aligned}$$

Sixth example ($\mathcal{FL}_0$): A sheepdog is a dog that among other things, guards only something.

$$\begin{aligned} Sheepdog &\equiv Dog \sqcap \forall guard.\top \\ Sheepdog^{\mathcal{I}} &= \{a \in Dog^{\mathcal{I}}\} \cap \{a \in \Delta^{\mathcal{I}} | \forall b.(a,b) \in guard^{\mathcal{I}} \rightarrow b \in \top^{\mathcal{I}}\} \\ &= \{Laika, Wastl, Wicki, Rex\} \end{aligned}$$

Appendix C

Syntactic Difference

A new definition of difference operator was given in [BKT02]. For $C - D$, the idea is to remove all sub-descriptions from a given concept description C, which are either redundant within C or already present in D. The only difference to Teege's difference operator is that the minimum w.r.t. to a partial order $\prec_d$ is used instead of the maximum w.r.t. $\sqsubseteq$. Finally, note that the difference between C and D is not a priori uniquely determined.

Definition 25 (subdescription ordering) *Let C, D be $\mathcal{ALC}$-concept descriptions in $\mathcal{ALC}$-normal form, then C is a sub-description of D, written $C \preceq_d D$, iff:*

- *$C \equiv \bot$, or*
- *C is obtained from D by removing or replacing certain parts of D by the bottom-concept.*

As an example of a sub-description let us consider the following concept description:

$$C = P \sqcap P \sqcap \exists r.P \sqcap \exists r.(P \sqcap Q) \sqcap \forall r.P \sqcap \forall s.(P \sqcap \neg P).$$

A possible sub-description C',

$$C' = P \sqcap \exists r.Q \sqcap \forall r.P \sqcap \forall s.\bot$$

is obtained from C as follows. Eliminate one P and the existential restriction $\exists r.P$ on the top-level of C. In the sub-expression $\exists r.(P \sqcap Q)$ remove P. Finally, in the value restriction for s replace $P \sqcap \neg P$ by $\bot$.

Definition 26 (syntactic definition) *Let C be an $\mathcal{ALC}$-concept description and D and $\mathcal{ALE}$-concept description. The syntactic difference $C - D$ of C and D is defined as minimal (w.r.t. $\preceq_d$) $\mathcal{ALC}$-concept description E with $E \sqcap D = C \sqcap D$, writte:*

$$C - D = min_{\preceq_d}\{E \mid E \sqcap D \equiv C \sqcap D\}.$$