Diplomarbeit

Designing an Intelligent, User-Friendly Knowledge Sharing Platform with Semantic Web Technologies

Michael Engler
FH Furtwangen
Fachbereich Digital Medien (Digital Media)
Studiengang Medieninformatik (Computer Science in Media)
Supervised by Prof. Walter and Prof. Dr. Eisenbiegler

Rheinbach, February 25th, 2005
Abstract

Often knowledge management systems tend to fail because they lack a tight integration into work tasks and do not consider the organizational and cultural context of their users. This shows that a design for a knowledge management system needs to take into account the goals and motivations of its users. Employing usability engineering, a human-centered approach for designing software applications, allows to sensibly regard the organizational and cultural context during the design of a knowledge management application.

Sir Tim Berners-Lee vision of the Semantic Web provides an environment to publish information with well-defined meaning. Ontologies are seen as a key enabler for this Semantic Web because they allow the publication of well defined information in a formal way. The information stored in an ontology can be turned into knowledge when used in a new context. It is believed that ontologies are able to store and retrieve knowledge better than other approaches because they allow the formal semantic annotation of data. An important task for designing and using an ontology-based knowledge management system is to create and evolve an ontology. This task can be achieved with ontology engineering processes such as DILIGENT.

In this work two use cases are researched, which are defined for the digital library case study of the SEKT project carried out at BT. The use case “search & browse” explores an interaction design for a digital library, which uses ontologies to enhance the searching and browsing for resources. The second use case “Jottings” suggests an interaction design for a knowledge sharing application, which provides an advanced bookmark function, describing bookmarks by assigning well-defined topics. This function is used to research how users with normal IT skills can extend an ontology by interacting with a software application. The understanding how users extend ontologies is guided by the ontology engineering process DILIGENT.

---

1 from the English word “to jot” meaning making a short note
Acknowledgments

Special thanks to Ian Thurlow and Nick Kings, who were always willing to help and have also done the proof reading for this work. Thanks to David Alsmeyer, John Davies, Paul Warren, Alistair Duke, Sandra Stincic, from BT and all others of the Next Generation Web Research Lab at BT, who have helped and made my stay in Ipswich very pleasant. Thanks to Chirstoph Tempich for supervising the work and many important discussions, York Sure, and others from the Institute AIFB, who have made this work possible. Also thanks to Christopher Adl, Kai Broszat, Diana Cheng, Evgeni Kravets, Fay Owston, Otmar Seckinger, and Bernhard Seckinger. Finally thanks to Prof. Walter for suggesting to research the topic Semantic Web and supervising this work as well as Prof. Dr. Eisenbiegler for supervising this thesis. Research reported in this thesis has been financed by EU in the IST project SEKT (IST-2003-506826).

Contacts of partners:

York Sure
Institute AIFB
University of Karlsruhe
Englerstraße 11
D-76131 Karlsruhe
Germany

John Davies
BT Exact
Orion 5/12, Adastral Park
Ipswich IP5 3RE
UK

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Diplomarbeit selbständig und ohne unzulässige fremde Hilfe angefertigt habe. Alle verwendeten Quellen und Hilfsmittel sind angegeben.

Rheinbach, den 25.2.2005
## II Design & Evaluation

### 5 Proceeding Deriving Design

- 5.1 Research .......................................................... 47
- 5.2 Deriving Personas and Context Scenarios .................... 48
- 5.3 Design of Prototype and Evaluation .......................... 49
- 5.4 Remarks on the Proceeding ....................................... 49

### 6 Requirements Analysis: the BT Digital Library

- 6.1 Initial Implementation of the BT Digital Library ............ 52
- 6.2 General Requirements ............................................ 52
- 6.3 Personas ............................................................ 54
  - 6.3.1 Persona Daniel - A Typical Library User ................ 54
  - 6.3.2 Persona Joe - An Expert User .............................. 55
- 6.4 Context Scenarios on “Search and Browse” ................... 56
  - 6.4.1 Scenario “Finding a Specific Resource” ................. 56
  - 6.4.2 Scenario “Finding an Unspecific Resource on a Specific Topic” ............................. 57
  - 6.4.3 Scenario “Finding an Unspecific Resource on an Unspecific Topic” ...................... 58
- 6.5 Context Scenarios on “Jottings” ................................. 60
  - 6.5.1 Scenario “Add a Jotting” .................................. 60
  - 6.5.2 Scenario “Use a Jotting” .................................. 62
  - 6.5.3 Scenario “Edit a Jotting” ................................ 63
  - 6.5.4 Scenario “Remove a Jotting” ................................. 64
- 6.6 Context Scenarios on “Advanced Ontology Adaption” ........ 64
  - 6.6.1 Scenario “First Installation of Knowledge Sharing Application” ...................... 65
  - 6.6.2 Scenario “Advanced Adding a new Topic” ................ 66
  - 6.6.3 Scenario “Reorganizing the Knowledge Base” ............ 67
- 6.7 Conclusions and Summary ....................................... 68

### 7 Prototype Design

- 7.1 Description of the Prototype “Search & Browse” ............. 69
  - 7.1.1 General Description ......................................... 69
  - 7.1.2 Expected Benefits ............................................ 70
  - 7.1.3 Detailed Description of Prototype ......................... 70
  - 7.1.4 Technical Issues ............................................. 78
  - 7.1.5 Potential Problems ........................................... 79
- 7.2 Description of the Prototype “Jotting” ......................... 80
  - 7.2.1 General Description ......................................... 80
  - 7.2.2 Expected Benefits ............................................ 82
  - 7.2.3 Detailed Description of Prototype ......................... 82
  - 7.2.4 Required Technologies ....................................... 92
  - 7.2.5 Potential Problems ........................................... 92
- 7.3 Rough Data Model ................................................ 93
- 7.4 Other design issues .............................................. 95
# Evaluation of Prototype

8.1 Results on the Use Case “Search & Browse” ........................................ 98
8.2 Results on the Use Case “Jottings” ...................................................... 101
  8.2.1 User comments ................................................................. 101
  8.2.2 Action Items ................................................................. 104
8.3 Results of the Questionnaire .............................................................. 105
8.4 Findings of the Evaluation ................................................................. 105

# Conclusion

III Conclusion ................................................. 107

9 Future Work and Conclusion ......................................................... 109
  9.1 Future Work ................................................................. 109
  9.2 Conclusion ................................................................. 110

# Appendix

IV Appendix ................................................. 113

A Additional Resources ......................................................... 125
  8.1 Information Search Tactics by Bates ........................................... 125
  8.2 Five Idea Tactics by Bates .................................................... 127
  8.3 Rough Data Model .......................................................... 128

B Artifacts of GUI-Design ......................................................... 129
  8.1 Paper & Pencil Prototype ..................................................... 129

C Artifacts of Evaluation ......................................................... 139
  8.1 Prototype of Evaluation: Search & Browse ................................... 139
  8.2 Prototype of Evaluation: Jottings .......................................... 144
  8.3 Questionnaire ................................................................. 151
Chapter 1

Introduction

In this thesis an application is developed for accessing digital resources (such as electronic articles or electronic books) in a digital library using newest Internet technology. In addition to accessing digital resources the application helps to organize digital resources. The resources are categorized with different topics, thus making it possible to retrieve one resource from different categories. Reaching resources from different categories breaks our usual understanding of how to file things because in the physical world we can only file a resource in one category viz in one file. The data in the application is organized in a structure called ontology, helping to search and organize information unambiguously in the correct context.

More precisely is the objective of this work to create an interaction design for the two use cases “search & browse” and “Jottings” of the digital library at BT\textsuperscript{1}, as part of a case study. The task is to provide a vision how the functionalities of the two use cases could be implemented using Semantic Web technologies. The use cases are developed as part of the SEKT project\textsuperscript{2}, which is partly funded by the European Commission and aims to enhance technology-based knowledge management.

The use case “search & browse” aims to answer the question of “How can we help users finding information better by using ontologies?” The outcome of the interaction design has been used in a deliverable for the SEKT project (see [13]).

The “Jottings” use case aims to answer the question of “how an interface looks like supporting the DILIGENT process”, thus supporting the local adaption of an ontology. The DILIGENT process is an ontology engineering process currently developed at the Institute AIFB at the University of Karlsruhe\textsuperscript{3}. The main goals of the DILIGENT process are to support ontology engineering in a distributed, evolving environment, possibly requiring little help of ontology engineers by providing a detailed description of the process steps. The prototype described in this work shows how an application can support the local adaption step of the DILIGENT process.

As a prerequisite this work explains the concepts of ontologies and their connection to the Semantic Web, an idea coined by Sir Tim Berners-Lee allowing agents to understand and process data available on the Internet. Ontologies are seen as a key concept for the Semantic Web as they are able to represent data in a semantically enriched way, in contrast to the data retrievable from the Internet of today. Ontolo-
1. Introduction

gies are a concept from the artificial intelligence community for storing knowledge. This semantically enriched data representation with ontologies can also be used for knowledge management systems, which are IT systems with the goal to manage knowledge of organizations.

Knowledge management, as a method of enhancing the way how knowledge is used in organizations, has to put humans in the center of its activity. In the past many knowledge management systems have been designed and implemented. Some of these systems failed because they did not take into account how humans would use the application. To avoid this shortcoming usability engineering is used to provide a sound understanding of the future users' work and culture.

The proposed prototype for the use cases “Jottings” and “search & browse” focuses on describing the interaction of a user with basic ontology-based functionality. Additionally some scenarios describe advanced features of an ontology-based knowledge management system, which can be used as a starting point to design a more sophisticated solution.

1.1 Structure of Thesis

This work is divided into three parts. The first part describes the theoretical background in the chapters 2, 3, and 4. The second part of the thesis proposes a interaction design for the two use cases “search & browse” and “Jottings” and provides an evaluation of this design. The part consists of the chapters 5, 6, 7, and 8. The last part (chapter 9) proposes future work and draws a conclusion.

Chapter 2 introduces knowledge management, which provides the overall background for this work. Then the idea of Semantic Web with its key enabler ontologies is introduced in chapter 3, providing an understanding of the underlying technology for the prototype. In chapter 4 Goal-Directed Design as a methodology from the usability community is visited, a model for classifying the goals of search engine users, and an overview of important search engine user interfaces is provided. Goal-Directed Design was used as a guideline for designing the prototype described in the second part.

Chapter 5 explains the proceedings followed during the design of the prototype. The requirements analysis was done using context scenarios, which are listed in chapter 6 together with some additional requirements. The prototype of the interaction design is illustrated with screenshots and descriptions in chapter 7. In chapter 8 a description of how the prototype was evaluated is given, as well as the results of this evaluation.

The work closes with chapter 9, which presents how the described prototype could be improved in the future, summarizes the results of this thesis, and draws a conclusion.

1.2 Scope of Thesis

This work proposed an interaction design for an ontology-based knowledge sharing application. Interaction design is a method form usability engineering, which uses techniques such as personas to foremost create a design putting humans in the center of every activity [27]. The approach seems to be especially suited for knowledge sharing applications as they need to deal to a large extend with human issues,
creating the need to focus on humans. The design shown is a good first sketch for an knowledge sharing application. For implementing the application it would be necessary to refine the design, e.g. with the creation of a better visual design.

This work does not provide a technical application design in the sense of e.g. object-oriented analysis and design. Some technical issues are brought forward and ideas for realization are made. These can only serve as input for an actual application design process.

Nevertheless the interaction design proposes some principle ideas, which enable a user with intermediate IT-skills to make use of an ontology for searching and browsing as well as classifying URLs and creating new ontological concepts.
Part I

Theory
Chapter 2

Knowledge Management

This first chapter describing underlying and related theory introduces knowledge management, which is a management approach for creating, distributing, sharing, and using knowledge in organizations. Knowledge management has been a topic of discussion and a buzz word for several years. As often many of the promises turned out to be exaggerating the usefulness of knowledge management. Nevertheless business leaders are realizing the fact that their companies uniqueness or value relies heavily on the knowledge the company has in developing and producing goods or providing services. Especially companies providing services have a keen interest in knowledge as their employees rely on applying knowledge a customer cannot access otherwise. This holds especially true for major consulting and IT-related companies. Another indicator for the importance of the topic is the research effort of creating a knowledge-based economy by the European Union, as stated by the European Commission:

"Research into Information Society Technologies (IST) is the largest priority in the EU’s Sixth Framework Programme (2002-2006), reflecting the importance of these technologies in achieving the Lisbon goal of turning the EU into the world’s most competitive, dynamic knowledge-based economy by 2010."[10]

Although knowledge management as such is mainly a concept which relates to cultural and organizational issues, it can be supported using information technology. Such technology support can range from groupware solutions and cooperate intranets to systems employing ideas from artificial intelligence like case-based reasoning, knowledge-based systems and so forth. The here presented application is part of the approach to use artificial intelligence in knowledge management. The expectations of using artificial intelligence though are somewhat weaker than some years ago.

2.1 Introduction to Knowledge Management

One of the main problems trying to explain knowledge management is due to the lack of an easy and clear definition of knowledge. Therefore the literature presents often the three related concepts of data, information and knowledge [28], [45]. Davenport
2. Knowledge Management

and Prusak interpret data as signs and information as a message containing signs. They define knowledge as:

“Knowledge is a fluid mix of framed experience, values, contextual information, and expert insight that provides a framework for evaluation and incorporating new experiences and information. It originates and is applied in the minds of knowers. In organizations, it often becomes embedded not only in documents or repositories but also in organizational routines, processes, practices, and norms.” [28]

This definition shows how complex the topic is. Knowledge can have very different appearances and is not “stable” but fluid. Implicitly this definition shows also that knowledge cannot only be explicit (e.g. captured in a document), but also tacit, and therefore in the minds of people. This shows how knowledge management needs to deal foremost with cultural and organizational issues. O’Leary and Studer state that experts at a “Dagstuhl Seminar on Knowledge Management” estimated that no more than 10-30 % of knowledge management can be covered by IT support [54].

As an understanding for knowledge management the definition of Abecker and van Elst [12] following Eppler is introduced. According to them Knowledge management can be defined as a:

- “systematic approach (with background in information technology, human resources, strategy, and organizational behavior)
- that views implicit [or tacit] and explicit knowledge as a key strategic resource, and
- aims at improving the handling of knowledge at the individual, team, organization, and inter-organizational level
- in order to improve innovation, quality, cost-effectiveness and time-to-market.” [33]

This definition shows that knowledge management aims at raising the competitiveness of companies by trying to enhance the way their employees use tacit and explicit knowledge within the company.

Important in knowledge management are the employees of companies (thus knowledge management is mainly concerned with cultural and organizational issues). Drucker has introduced a term for employees working in a knowledge-based economy: The knowledge worker.

“The manual worker is yesterday [...] The basic capital resource, the fundamental investment, but also the cost centre for a developed economy is the knowledge worker who puts to work what he has learned in systematic education, that is, concepts, ideas and theories rather than the man who puts to work manual skill or muscle.” [39]

In [28] Davenport and Prusak provide a market-based model how these knowledge workers share their knowledge. They identify three different types of how knowledge workers share their knowledge. They are called seller, buyer and broker. The types are rather straightforward to understand. A seller is a knowledge
worker, who can offer knowledge to others, a buyer is somebody, who is looking for knowledge he needs to do his job, and a broker is somebody, who connects with many sellers and buyers allowing them to meet and share their knowledge. Davenport and Prusak describe the currency as reciprocity, repute, and altruism (ordered from greatest to least). Important for knowledge sharing applications is to keep this model describing the motivations of knowledge workers in mind because otherwise the design and deployment of an knowledge sharing application in the work environment can easily fail. This market-based model helps to understand users’ motivations, thus explaining why a tight integration in users’ work tasks is so important.

2.2 Knowledge Management Systems

As stated before knowledge management mainly concerns cultural and organizational aspects. This does not imply that it is impossible to support knowledge management with IT-systems. Especially large companies and companies, which are distributed over several locations, can benefit from knowledge management systems. Knowledge management systems are “enabling technologies for an effective and efficient knowledge management” [47]. The accessibility (e.g. over time) of knowledge can be improved by knowledge management systems as well as the way of communicating across physical boundaries, using cooperate internets or groupware systems. Knowledge management systems might provide means to cross cultural or organizational boundaries, but to be successful there needs to be also a cultural or organizational change adopting those tools. Examples for this could be expert locators, e-learning, or best practice databases. An overview about current knowledge management systems can be found in [47] and [55].

A more specific kind of tools are knowledge sharing applications. In the use case “Jottings” a knowledge sharing application will be described. The aim to enable users of the BT Digital Library to create semantically annotated bookmarks (Jottings), which are shared by default with other users. Riedl describes in his work “Some Critical Remarks in Favour of IT-Based Knowledge Management” the function of knowledge sharing applications as follows.

“There are various types of tools [or applications] supporting knowledge sharing processes. Their function is fivefold. They enable us to document and structure knowledge; they enable us to transfer structured or unstructured knowledge to a dedicated partner; they enable us to store and retrieve structured knowledge; they enable us to publish and retrieve semi- and unstructured knowledge, and they enable to observe the behaviour of users. [...] What distinguishes knowledge management tools from data management tools is that they deal with data plus semantics, not with data alone.” [59]

The application proposed in this work will focus on four of these functions. The documentation and structuring of knowledge is done in the use case Jottings by assigning topics to resources. A Jotting can be sent to another user, thus sharing knowledge with a dedicated partner (see section 6.5 for both functions). The storage of knowledge can be done with the Jotting functionality described in section 6.5. The retrieval of knowledge can be done with the functionality as described in section 6.4 on the use case “search & browse”. Both use cases researched here allow to
2. Knowledge Management

observe the behavior of users. Only the function for publication and retrieval of unstructured or semi-structured knowledge is not covered in this work.

Beside intranets or groupware tools, the artificial intelligence community has developed tools for knowledge sharing and management. The artificial intelligence community has quite a long experience in experimenting with knowledge. Knowledge was seen as a key enabler for artificial intelligence applications. Efforts were made such as expert systems trying to provide problem solving intelligence. Those systems inevitably fail, when they are applied in domains with a higher complexity of facts and rules. The artificial intelligence community came up with a different, less sophisticated approach though by using only a rudimentary set of rules in ontology-based systems. Those systems enable users to share the semantic understanding of data. Important for the new systems from the artificial intelligence community is that they do not rely entirely on computing power, but allow a user to interact with the system, whenever no sufficient results can be achieve by machine processing.

2.2.1 Survey of relevant Knowledge Management Systems

In this section a short review of two knowledge management systems is provided, which played a role guiding the interaction design in this work. Some principles and ideas from these applications are used for the design of the prototype described in section 7. First a short overview of the tool del.icio.us [2] is given and then the Jottings functionality of the BT Digital Library is described.

The tool del.icio.us is described by its developers as social bookmark tool. The idea is that a community of users stores bookmarks in a database, allowing to view each others bookmarks. Everybody is invited to store new URL’s in the database annotated with self-defined tags. If bookmarks are stored by more than one person it is made explicit, as it is an indicator that the link might be interesting. Additionally a user can search the database to find URLs which are annotated by others. A messaging functionality allows to send questions and comments to other users. The idea of the application is very interesting as it is an intrinsic motivation for people to store bookmarks (they want to remember the bookmarks themselves), but they also add knowledge (about good websites) to the community of all users (see also figure 2.1).

The Jotting functionality of the Digital Library is very similar to del.icio.us. Compared to del.icio.us it omits the community aspect. This Jottings functionality just provides a way to store bookmarks centrally on a server, thus having them available on all web accessible digital devices using HTML. How a posting for a Digital Library Jotting can be done is shown in figure 2.2.

Other systems and application relevant to this work will be surveyed in chapter 3.6 and 4.3.
2. Knowledge Management

Figure 2.1: Posting a URL using del.icio.us.

Figure 2.2: The Jotting functionality of BT Digital Library
Chapter 3

Knowledge Management using Ontologies and the Semantic Web

This chapter introduces the concept of ontologies and the Semantic Web, which relies on ontologies. In a later part of this chapter a process for developing ontologies (ontology engineering) is described. This process will be supported by the use case Jottings. What kind of related technologies are available, and how to search the Semantic Web follows. Finally relevant application are introduced, mainly developed by the research community.

3.1 Introduction to the Semantic Web

The Semantic Web is an idea introduced by the inventor of the Internet architecture as of today, Sir Tim Berners-Lee. In 2001 Berners-Lee introduced in a famous article in the Scientific American [22] his idea of making the web not only a source for information that can be read by humans, but also enabling machines to process semantics of information provided. In this section principles of the Semantic Web are described, followed by some common visionary scenarios, how the Semantic Web could be used.

In their article “W3C Semantic Web Activity” [40] Koivunen and Miller describe six principle ideas the Semantic Web is based on. These principles are summarized in the following:

Everything can be identified by URI’s The normal web technology already uses Unique Resource Identifiers (URI, see also [21]) to address documents. The idea of URIs is that every document has a unique address for retrieval. In the Semantic Web such a URI can be assigned not only to documents, but people or any object in the world, thus making it addressable for machines. For example the FH Furtwangen could be identified by “http://www.fh-furtwangen.de”. This is a pragmatic approach, but generates some problems. The URI “http://www.fh-furtwangen.de” would then point to the university as institution as well as to the website of the university. With more metadata it would be possible to describe both entities separately.
3. Knowledge Management using Ontologies and the Semantic Web

Resources and links can have types On the Internet as of today links do not have types. A human can understand easily if a webpage is an invoice, political news, or a scientific article. Computers have problems doing so. Therefore it is possible in the Semantic Web to give links and resources types. A type for a link could be “dc:creator” or “dc:title” and for a resource “my:engler” or “my:highstreet”.

Partial information is tolerated In the web of today we live with the 404 Error message, indicating that a webpage has been removed. The same thing needs to apply to Semantic Web resources. Often it is enough to get partial information. Tolerating partial information though creates challenges for the termination of inference engines and query algorithms. This is yet an unsolved problem.

There is no need for absolute truth Not every information needs to be true. More pragmatically one could talk about trustworthiness, which needs be inferred by applications processing data from the Semantic Web. The applications decide when to trust which statements by using the context of a statement. For example if his manager had told James to upload a document for him, a Semantic Web application could permit James to upload the document even though he has not the password required. The application would infer that James is acting on behalf of his manager, who has the password.

Evolution is supported Similar concepts are often developed by different groups of people at different times or the understanding of a certain domain evolves so that the vocabulary also changes. The Semantic Web can expand as human understanding expands. It can resolve ambiguities and clarify inconsistencies, as well as translation in the case of different vocabulary. Old resource do not need to change because new information has been added.

Minimalist design The Semantic Web as such only specifies possibilities how to express something. It does not provide a specific way of how to express it. The Semantic Web for example does not provide a specific way how to express that a person is the author of a resource. Using Semantic Web recommendations one could use several ways to express that a person is the author of a resource. These more specific definitions are provided for example by the above mentioned Dublin Core standard.

The technical architecture of the Semantic Web can be seen in figure 3.1. The expressiveness of the layers rises from bottom to top. The higher a layer is located the more powerful expressions can be constructed. The bottom layer consisting of Unicode and URI makes sure that an international character set is being used as well as a unified approach to addressing resources. The “XML + Namespaces + xmlschema” layer ensures compatibility to other standards and applications using other XML based standards. “The RDF + rdfschema” layer is the first data layer, which allows to create statements about objects and vocabularies that can be

---

1 Resources in this case can be anything that is addressable by a URI e.g. a webpage, a picture, etc.
2 Both link types are taken from the Dublin Core Standard (namespace is dc) [9] and all examples are Qualified Names as defined in [5]). The namespace “my” is an indication that any URI can be used as a resource, even though this might not always be desired.
3. Knowledge Management using Ontologies and the Semantic Web

Figure 3.1: The figure shows the architecture layers of the semantic web. [40]

referred to by URIs. This layer allows to give types to resources and links. The ontology layer allows to create RDF vocabularies with more constrains. This allows to support more sophisticated reasoning about these vocabularies and their resources. The “Digital Signature” layer allows to detect if a statement had been altered. The standards described so far are mostly already recommendations of the W3C. The top layers such as logic, proof and trust are researched as of now. Some application demonstrating simple features are developed. The logic layer enables to write rules, which the proof layer executes. Together the trust layer and the proof layer decide to trust or reject the proof provided. (Following [40])

After reviewing some basic principles and the architecture of the Semantic Web, some visionary application scenarios of the Semantic Web are described in the following. In his article in the Scientific American Tim Berners-Lee described his vision of the Semantic Web as an environment where agents fulfill complex requests for users. Berners-Lee states an example which describes how a software agent makes appointments for medical treatment for a relative of a person. Therefore the agent needs to find a facility that is supported by the health insurance, but near to the home of the person making the request. Additionally the agent schedules several appointments by using information from the schedule of the person, as well information from the health insurance and information from the health care facility.

Another example application for the Semantic Web are Semantic Web Services. These Semantic Web Services provide a formal semantic description of their functionality. Right now the input and output of Web Services are described by WSDL. This description defines the syntax of the input and output of a Web Service, but omits to define the semantics in a machine processable way. Semantic Web Services aim to fill this gap, which might enable the dynamic binding of such services in the future. One could also imagine that an agent could create a new web service by using several different other Semantic Web Services (for more see [38][3]).

A third example for the applicability of the technology the Semantic Web provides is data integration. Lakshmanan and Sadri describe in [43] a framework to provide wide-scale data integration. They envision that data sources, which are interesting for the participation in data and application sharing should be enriched with semantic declarations. These semantic declarations provide a semantic view on the information contents. Ontologies could exist to relate terminologies used across

[3]Additionally information from Nick Kings has been used in this paragraph.
different communities of an application domain or even different application domains. With this approach queries can be constructed using several different data sources by consulting ontology servers providing common ontologies. Interoperability systems would be responsible for processing and optimizing these queries and would carry out this task in accordance to ontology servers providing the common terminologies. Lakshmanan and Sadri do not expect one common ontology for a given application, but rather that within one application domain there will be a limited number of widely used ontologies.

As stated earlier the Semantic Web is still a research topic. Many issues still need to be resolved to put Berners-Lee vision into practice. One examples is to create a common query language for RDF. Several approaches are being explored such as [30], but there is no standardized way of how to formulate queries such as in SQL for relational databases. Some other issues are the expression of classes as property values [11] and how to present n-ary relations [8] on the Semantic Web.

### 3.2 Introduction to Ontologies

Here a short account of the historical development of ontologies is provided, a definition is given what computer scientists mean when speaking about ontologies, and some examples of ontologies are stated. Then are following some usage scenarios, and finally potential problems with ontologies are discussed.

In the following we will give a short account of the historical development of ontologies as described in [63] by Staab and Studer. At first, in the early 1990s ontologies were used by the Knowledge Acquisition Community, which used ontologies for knowledge modeling. This resulted in the development of the CommonKADS methodology for knowledge engineering and management. In the later part of the 1990s it was recognized that “a conceptual, yet executable model of an application domain provides a significant added value for all kinds of application scenarios like knowledge management or eCommerce [...]” [63]. Ontologies became a major research topic with Tim Berners-Lee idea of the Semantic Web. Ontologies are seen as a key enabler to realize the Semantic Web.

This work uses a definition of ontology following Tom Gruber’s work in [36], which is summarized in [63]: “An ontology is a formal explicit specification of a shared conceptualization for a domain of interest.” To understand this definition some key concepts are discussed. Then it is shown that this definition has some rough edges according to Musen in [25]. In the definition the words “formal” and “explicit” mean that the ontology is written down in a formal way, indicating that the ontology is computer processable. “Shared” means in this context that a group of people has a common understanding of the ontology. It needs to be noticed that an ontology is in our case only for a certain domain. Problematic in Grubers definition is the understanding of “specification of a conceptualization”. The research community agrees that it includes taxonomic hierarchies of concepts and attributes of concepts. The research community does not agree though on how expressive constraints on such attributes and relationships between concepts and attributes (in any kind of combination) should be, either allowing only simple role restrictions or more complex logic. This is reflected also in the Web Ontology Language from W3C (see section 3.4.2), which has three layers providing different levels of expressiveness. It can be summarized that an ontology is an enumeration of concepts of an application domain. Usually properties and relations can be expressed. But what
kind of constrains can be placed on relations and properties is not agreed upon (following Musen in [25]).

The theoretical basis for ontologies are Description Logics and F-Logic. F-Logic (F refers to frame) is a frame-based language, which combines the advantages of the high-level approach of a frame-based system with the expressiveness, the compact syntax, and the well defined semantics from logics. The original features of F-logic include signatures, object identity, complex objects, methods, classes and inheritance. F-Logic accounts in a clean and declarative fashion for most of the structural aspects of frame-based and object-oriented languages. OWL lite, the lowest layer of OWL, fully represents F-Logic (following [15]).

The name description logics comes from the idea of describing domains with concept descriptions and using formal logic-based semantics. Description logics are more expressive than F-Logic. One can express unions, intersections and complement and more powerful value restrictions with description logics in comparison to F-logic. Description logics allow the subsumption of classes and therefore are better suited for the actual process of modeling ontologies. The additional features of description logics rise the computational complexity, thus description logics are not well suited for ontology-based applications using large sets of instances (from [52], [15] and [18]).

An example for an ontology are relationships between relatives. On one hand there is the relationship of parents to children, and on the other hand there are such relation as aunt or uncle, nice or nephew, and cousin. Other examples can be thesauri such as the one used for the Inspec data base [7]. The thesaurus of the database provides a controlled vocabulary to classify the articles within the database. The terms in the vocabulary are ordered hierarchically. Additionally related terms are listed. This creates an ontology of terms using the two relations “is subtopic of” and “is related to”. To be more concrete the term “Telecommunications Security” has the broader terms “Security” and “Telecommunication” and has as a narrower term “Message Authentication”. An example for a term related to “Telecommunications Security” is “Security of Data”\(^4\) (see figure 3.2).

Ontologies might be laborious to construct and it might also be hard to reach an agreement on how to conceptualize a domain. More critical according to Ellman is “developing the content to populate it [the ontology] in sufficient depth to make it useful in a real application.” [25]. Ellman refers to the problem of creating and adding enough instances (or information) to make an ontology useful. As of today there are approaches for creating metadata using natural language processing, but these have the problem that they cannot classify all documents correctly, thus making it necessary that a human needs to correct the errors. For the prototype design this only applies to the use case “Jottings” because the other resources of the BT Digital Library are already classified. For example the before mentioned Inspec database classifies the contained documents already in a high-quality fashion, which allows to easily add those documents to the ontology.

Another critical point in using ontologies is that ontologies are hard to maintain. As the real world is changing so are the domains an ontology represents. Therefore

\(^4\)The example is from an older version of the thesaurus and due to changes might not be found in the thesaurus of 2004. Important here is that the structure of the thesaurus has not changed, thus the example still makes its point.
the ontology needs to be updated to still represent the conceptualization of a certain domain. This problem can be tackled by process-driven ontology evolution. According to Stojanovic et al. “ontology evolution is the timely adaptation of an ontology to changed business requirements, to trends in ontology instances and patterns of usage of the ontology-based application as well as the consistent management/propagation of theses changes to dependent elements. [64]”. The section on ontology engineering will introduce a process, which is designed to help solving the problem of evolving ontologies. Another solution, though not yet applicable because very much a research topic, is introduced by Staab in [62]. He proposes “emergent semantics”, which is the idea to observe the interaction between humans and computers to infer how semantics are evolving. According to Staab the advantages are threefold, thus (i) it might solve the above described problem of creating instances for ontologies, which is called the “knowledge acquisition bottleneck”, (ii) the observation of human-computer interaction does not require additional human effort, letting semantic structures evolve on their own, (iii) a new basis for understanding natural language can be created combining logical structure of language and context.

Ellman summarizes in [62]: “The challenge in representing knowledge with ontologies don’t seem to lie in issues of representational adequacy or underlying formalism. Rather, we find them in the mechanics of integration with existing systems and the design or acquisition of content that’s appropriate to the required function in the information interface.”

### 3.3 Ontology Engineering

Ontology engineering is a methodology, which is a specialized form of knowledge engineering. Following Schreiber et al. knowledge engineering provides theories,

---

5 This is a strong simplification of Staab's point. The interested reader is referred to his article “Emergent Semantics” [62]
methods, and techniques to analyze and engineer knowledge in form of a scientific methodology [60]. Knowledge engineering and more specific the knowledge engineering methodology CommonKADS [60] has a somewhat broader scope compared to ontology engineering, as it not only deals with formalizing knowledge in a model, but also provides a whole framework for projects developing knowledge management systems with guidance for feasibility studies, application design, coding etc. (from [60]). Ontology engineering in contrast focuses on designing a formal knowledge model in form of an ontology. Creating an ontology is a laborious, complex task, which has high requirements on the quality of the output. For example an ontology needs to be correct and consistent for reasoning to provide valid answers. Additionally the aspect of sharing the conceptualization needs to be fulfilled, requiring to reach an agreement on the ontology design. Therefore it is necessary to provide a methodology supporting the design of an ontology with theories, methods and techniques. An application using ontologies, such as the prototype developed later, needs to employ ontology engineering to manage the underlying ontology in a professional way. Here the ontology engineering process DILIGENT will be used as part of a research effort to find out how the step local adaption of the DILIGENT process can supported by a knowledge sharing application. In the remainder of this section will follow a description of the On-To-Knowledge methodology, which serves as an example for the shortcomings of many ontology engineering methods. Then a description of the DILIGENT process is given. Finally supporting technologies and ontology engineering tools are briefly visited.

By now the scientific community has suggested a number of ontology engineering methods. In this place we will only mention the On-To-Knowledge methodology [65] as an example for a more traditional ontology engineering methodology, which has certain shortcomings especially for being used in a distributed, dynamic environment such as the Semantic Web. On-To-Knowledge subdivides knowledge engineering in two processes, which are orthogonal to each other. The first one is called the “Knowledge Meta Process” and focuses on introducing and maintaining a knowledge management system. The second one, called “Knowledge Process”, addresses the handling of an already deployed and running knowledge management system. In [65] Sure et al. describe the Knowledge Meta Process as consisting of the steps feasibility study, kickoff, refinement, evaluation, and application & evolution, whereas the last three steps are performed usually in an iterative cycle. The resulting artifacts of the steps in the same order are filled out worksheets from CommonKADS, an ontology requirements specification document and a semi-formal description of the ontology, the target ontology, an evaluated ontology, and an evolved ontology. The knowledge process is not described in great detail. It basically consists of the steps knowledge creation and import, capturing of knowledge items, retrieval and access of knowledge, and use of knowledge. Notice should be given to the fact that On-To-Knowledge describes how ontology engineering tools support different process steps. Sure et al. describe the shortcomings in [65] as follows: “Still, there are many open issues to solve, e.g. how to handle a distributed process of emerging and aligned ontologies that is likely to be the scenario in the semantic web.” The successor of this process, trying to solve these shortcomings is named DILIGENT.

### 3.3.1 The DILIGENT Process

The DILIGENT process mainly introduces a methodology for distributed ontology engineering and the use of ontologies in a distributed, dynamically evolving
application. This is supported by allowing users to modify their copy of a shared ontology, thus adapting it to their own needs. The process then supports to reach an agreement again on how the shared ontology should conceptualize the given domain. DILIGENT tries to support ontology engineering emphasizing a very formal approach to the presentation of arguments. The process contains the five steps (i) build, (ii) local adaption, (iii) analysis, (iv) revision, and (v) local update. A process a visualization can be found in figure 3.3. A closer look at each step follows:

Figure 3.3: The DILIGENT process

- **Build** The build step requires the design of an initial ontology. Different stakeholders, with usually different purposes and needs build together an initial ontology. As especially the distributed engineering of ontologies is supported, those stakeholder do not need to be at the same geographical location during the ontology engineering. Important is that employed ontology engineering tools need to provide online support. Ontology engineers create this initial ontology with the help of domain experts and future users.

- **Local adaption** This step will be described in more detail because the scenarios on “Jottings” explicitly support the local adaption step of the DILIGENT process. This step is the “use” step of the initially build and deployed ontology. The main idea in DILIGENT is that users are now able to modify their copy...
of the shared ontology, but these changes are not propagated to other users. In
the later steps the ontologies are being aligned again. The actions a user needs
to perform are to understand the shared ontology, identify the commonalities
between own and shared conceptualization, map equivalent conceptualizations
of different actors, identify missing conceptualizations, change conceptualiza-
tions, and finally organize local knowledge according to the conceptualizations.
The steps are now described in more detail:

– **Understand shared ontology** In this step the user learns, which con-
cepts are available, where they are located and how they relate to each
other in the shared ontology. Important for large ontologies might be
to provide a context related view, thus avoiding to overwhelm the user
with the ontology. This task can be supported with such technologies as
text classification methods, natural language processing, and ontology
learning.

– **Identify commonalities between own and shared conceptualiza-
tion**: Now the user identifies commonalities between his own conceptu-
alizations and the conceptualizations provided by the shared ontology.
This can lead to different forms of conflict⁶. To support the user in this
step formal conceptualizations on the local machine (e.g. file structures)
can be used to create mappings between the locally available conceptual-
ization and the shared ontology. Additionally ontology learning methods
can be used to identify concepts and relations based on text available on
the machine of a user.

– **Map equivalent conceptualizations of different actors** This step
is necessary for the application to use the identified commonalities in
the case of a common concept that is named differently. Mappings are
important because they keep the original structure of the shared ontology.
Changes must be traced to go back to the previous version.

– **Identify missing conceptualizations** The user also needs to identify
missing conceptualizations. To find the missing concepts ontologies of
other users (if available) can serve as an suggestion.

– **Add missing conceptualizations** The user needs to introduce the
changes to the ontology, which is rather a challenge of an appropriate
interaction design than a technical challenge. Automated methods are
not reliably enough for this task. Therefore the user needs to actively
support the application. The support needs to be done in an undisturbing
way for the user. The user should according to the process description
provide reasons why needs to add a new concept to the ontology.

– **Organize local knowledge according to ontology** In this step the
user adds instances to the ontology, thus populating the ontology with
resources and adding to the shared knowledge base. Text classification
and natural language processing can be used to support this step.

⁶Those can be:

*Consensus*: the same terminology is used to describe the same concept

*Correspondence*: different terminology is used to describe the same concepts

*Conflict*: the same terminology is used to describe different concepts

*Contrast*: different terminology is used to describe different concepts (see [61])
3. Knowledge Management using Ontologies and the Semantic Web

- **Analysis** The analysis is carried out by a board, where domain experts, user representatives, and an ontology engineer decide, which changes in a new version of the shared ontology should be introduced. The input for the analysis comes from the collected changes introduced to the shared ontology by users. The provided reasons for the changes should help the board to decide, which changes should be introduced in the shared ontology. Important to notice is that not all changes will be introduced into the shared ontology, thus keeping the shared ontology consistent and usable. The results are a list of changes, which should be applied to the shared ontology. Actions that need to be taken to complete this step are to gather locally updated ontologies and corresponding arguments, analyze the introduced changes and identify changes relevant for all users.

- **Revision** The revision of the shared ontology is done by the ontology engineer and the domain expert. The ontology engineer might correct wrongly placed concepts and the domain experts might introduce additional concepts and relations not requested by users. The output of this phase is a formalized, well documented, changed, shared ontology. Actions that need to be done are the formalization of the requested changes, aggregation of arguments, and documentation.

- **Local update** In the local update step the users receive the changed shared ontology and the documentation of changes. The user can now decide, which changes that have been made to the shared ontology he wants to adopt. The user is not forced to adapt to all changes of the ontology. It is one of the main points of the DILIGENT process to allow the user to represent his world view in his copy of the share ontology. Therefore it would not make sense to force a user to accept all changes of the shared ontology. Actions in this step are distribution of the ontology to all actors, tagging of the updated version, inclusion of the updated version, and update of local adoptions which are not included in the shared ontology. Now the process starts again with the local adaption step.

Every argument brought forward to change the ontology should be classified in DILIGENT according to the Rhetorical Structure Theory. Only a restricted set of arguments is seen as valid arguments in DILIGENT. Sure et al. hope with this restriction to allow a more effective and efficient discussion. It is questionable if users will be willing and able to give reasons for every change of the ontology they make. It is not a user's goal to create a good ontology, but rather to get their normal work tasks done.

In the now existing process descriptions of DILIGENT only the last three actions of the local adaption step are seen to be preformed in a cyclic manner. This might be an oversimplification of the first three actions, especially if the application making use of the DILIGENT process should be embedded seamlessly in the work environment of users. It might show that all actions are performed together in a more or less cyclic manner. This might hold true because the user is not familiar with the ontology when he first uses the ontology to organize his knowledge and he might not want (or have the time) to review the ontology as a whole. This is especially the case when using a large ontology. A user might not be able to understand this large ontology fully at all, but only the parts he is really interested in. The application design in this work follows the later idea, that all six actions are rather
mixed together because the user explores the ontology only bit by bit learning new things when he feels the need to do so. This means that a normal user might never explore the ontology as a whole.

Important for every ontology engineering process is to integrate available tools as closely as possible to support ontology engineers in their task of creating ontologies. The DILIGENT process uses for this the tool OntoEdit\footnote{OntoEdit is a development of ontoprise. See also http://www.ontoprise.de/}. OntoEdit has been used in one of the case studies using the DILIGENT process, which is described in [56]. An overview over ontology engineering environments can be found in [51]. Other ontology engineering processes are described in [65].

The DILIGENT process is still being developed and holds some important promises for the future. Particularly the DILIGENT process addresses issues regarding the design of ontologies for the Semantic Web, which would provide a very similar environment as the DILIGENT process, thus being a distributed and evolving. The application designed in this work is among the first efforts of developing a tool that is especially designed in regard of the DILIGENT process.

### 3.3.2 Terms Related to Ontology Engineering

Before moving on some terms related to ontologies and ontology engineering need to be explained. Ontology Learning refers to a semi-automatic machine learning approach for constructing ontologies. Semi-automatic means that ontology engineering is a combination of machine learning and ontology engineering. For ontology learning machine learning algorithms such as lexical entry and concept extraction, clustering, classification, and lexico-syntactic pattern are used. Maedche and Staab regard “ontology learning as an approach that may greatly facilitate the construction of ontologies by the ontology engineer” [46].

Merging and mapping ontologies is an important task to ensure the interoperability of different ontologies. It is a laborious and error-prone task to compare two ontologies because one must find all similar concepts in the two ontologies, determine the similarities and either remove the overlapping concepts in the source ontologies or record the mapping. For an efficient ontology mapping (leaving the two ontologies separate, but find commonalities) or ontology merging (creating one ontology containing all concepts of both ontologies) it is necessary to provide support through automating parts of the task [53].

Ontology evaluation refers to the idea to test the quality of a constructed ontology. In [34] Gomez-Perez gives an introduction to the field. Up till now only the evaluation of taxonomies is possible. For other, more complex ontologies a method is not available.
3. Knowledge Management using Ontologies and the Semantic Web

3.4 Semantic Web Technologies for Knowledge Management

This section introduced the two W3C recommendations “Resource Description Framework (RDF)” and “Web Ontology Language (OWL)”. RDF has been revised within the effort on providing recommendations for the Semantic Web. OWL is a new recommendation, also developed with in the Semantic Web effort\(^8\) of the W3C.

3.4.1 The Resource Description Framework (RDF)

The Resource Description Framework (RDF) is a W3C recommendation\(^9\), which was defined as a language for metadata about web resources such as webpages or pictures. RDF has a representation formalism, which allows it to be written in XML \[^3\]. RDF as such describes resources with named properties having values. RDF schema allows the definition of vocabularies for RDF, which describes properties, classes of resources and relationships between them. RDF follows a main web principle that anybody should be able to say anything about anything\(^10\). As RDF is distributed it allows in contrast to traditional object-oriented or frame-based systems the introduction of new properties to existing classes. Traditional centralized systems would need the cooperation of the owner or the declaration of a subclass to add a property. Therefore RDF is property centric, which means that it enables the definition of properties and their use for the description of resources without the need for changing the original class or creating a new subclass. RDF schema language constructs allow the definition of classes and subclasses, properties and subproperties, domain and range\(^11\) constraints as well as built in vocabulary for containers, collections and reification\(^12\).

Finally it is important to recognize that RDF has formal semantics. This has two reasons. Firstly it makes the RDF specification more precise and avoids implementations to differ too much in scope. Secondly it prevents different extensions of RDF (such as OWL) to define their own semantics, possibly differing from each other (from \[^50\] and \[^48\]).

3.4.2 The Web Ontology Language (OWL)

The Web Ontology Language\(^13\) extends RDF to reach higher expressiveness. The starting point for OWL was an European and American joint initiative called DAML+OIL (see \[^6\]) for defining a powerful ontology language. To understand how OWL and RDF schema relate to each other it is necessary to know, which requirements an ontology language with higher expressiveness has and what limitations RDF schema has.

\(^8\)See also http://www.w3.org/2001/sw


\(^10\)It needs to be noted that some statements cannot be expressed with RDF.

\(^11\)Domain indicates that if an instance has a particular property it is always member of a designated class. Range indicates that values of a particular property are of a designated class. See section 5.2. of \[^48\] for more information.

\(^12\)Reification statements are a built in mechanism of RDF schema to describe other RDF statements. For example it could be useful to describe who created a statement and when it was created.

One requirement is that the ontology language needs a well defined syntax so that it is machine processable. Precise formal semantics are needed to give no room for subjective interpretation and enabling to reason about knowledge. Using formal semantics allows to reason about class membership, equivalence of classes, consistency, and classification. Reasoning support is important for ontologies because it allows consistency checks, checks for unintended relations between classes, and automated classification of instances, thus enabling machine support constructing and populating ontologies.

Some of the limitations of RDF are listed in the following limitations: One cannot declare properties that have a range restriction for some classes only. In RDF schema the range restriction is always applied to all classes. RDF schema does not allow to state explicitly that classes are disjoint and also boolean combinations of classes are not possible. Cardinality restrictions are not possible as well as special characteristics of properties such as being transitive, unique, or inverse compared to another property. It is problematic to just extend RDF with the constructs described above because the modeling primitives rdfs:Class (the class of all classes) and rdf:Property (the class of all properties) are very expressive and may lead in combination with the constructs described above to uncontrollable computational properties.

To overcome the limitations and problems of RDF OWL was developed. To fulfill the diverging requirements of rich expressiveness and efficient reasoning support OWL is divided in three layers:

**OWL Full** allows all language constructs to be used in any kind of combination as long the result is a legal RDF document. It is fully compatible with RDF and RDF schema, but provides no complete reasoning support.

**OWL DL** introduces some constrains on the way how constructors from OWL and RDF can be used, which permits efficient reasoning support. It though looses full compatibility with RDF. The constraints are the following:

- **Vocabulary partitioning** meaning that any resource can only be either a class, a datatype, a datatype property, an object property, an individual, a data value or part of the built in vocabulary.

- **Explicit typing** means that the partition (as described above) needs to be explicitly stated by typing a resource.

- **Property separation** means that the set of object properties and datatype properties are disjoint. This is already included in the first statement.

- **No transitive cardinality restrictions** Cardinality restrictions cannot be applied to transitive properties or their subproperties.

- **Restricted anonymous classes** Anonymous classes can only be used in the domain and range of stating that a class is equivalent, disjoint or in the range of stating that a class is a subclass.

**OWL Lite** is easier to implement and easier to grasp for a users, but has of course less expressiveness. It is decidable with F-Logic. It additionally restricts the use of constructors stating an enumeration (owl:oneOf), being disjoint with another class (owl:disjointWith), being a union (owl:unionOf) or the complement of another class (owl:complementOf), or restricting a class of having a certain value (owl:hasValue). Cardinality statements can only be one or
zero and stating that a class is equivalent cannot be used with anonymous classes.

When developing an ontology, one should ask himself how powerful the expressiveness of the used ontology language needs to be and how much it is necessary to have reasoning support. Then one can decide on using a specific OWL layer. Finally it needs to be mentioned that OWL is fully upward compatible, but only to a certain extend downward compatible (from [16]).

3.5 Searching the Semantic Web

In this section some approaches on how to search in a semantically rich knowledge base are introduced\textsuperscript{14}. In principal key word searches can be improved by query enrichment (adding new terms or boolean connectors to a query) and term disambiguation (of homonyms\textsuperscript{15}). Another approach more directed toward browsing is to map keywords entered by a user to concepts of an ontology, thus allowing disambiguation and query reformulation. This approach is employed by Davies et al. in their application QuizRDF (see [30]). Disambiguation of query terms is used by de Luca and Nünberger as described in [44].

A typical search with QuizRDF starts of with a keyword search. Additionally to the result set the concepts used to describe all documents within the result set are retrieved and presented to the user. This allows a user to formulate a query against the ontology retrieving documents which are classified with the selected concept. Additionally the properties of this concept are retrieved as well as related concepts.

De Luca and Nünberger use MultiWordNet\textsuperscript{16} for disambiguating searches. They employ a search engine such as Google to search the web and if a word has different meanings (homophone) classify the results accordingly to the different meanings provided by MultiWordNet. This classification is then presented to the user.

Noteworthy is also the work of Goncalves et al. In [35] they claim that structured queries perform better than unstructured queries. Structured queries use only specific data fields, such as the title field or the author field. Thus they make use of annotations in comparison to full text searches, which treat all fields the same way. This supports the idea of using metadata (which is nothing different than the mentioned fields) for enhanced searching. Metadata provides of course nothing else but semantics, which can be described in an ontology. If the statement of Goncalves et al. holds true than using ontology-based applications for searching should improve search effectiveness.

3.6 Survey of Relevant Applications from the Research Community

In this section some tools are introduced developed by the research community to explore the possibilities of ontologies and the Semantic Web for knowledge management.

\textsuperscript{14}The term Semantic Web is avoided as the here introduced approaches are not necessary scalable to the Semantic Web.
\textsuperscript{15}Homonyms are words which have the same form (orthographic and phonetic) but unrelated meaning.
\textsuperscript{16}see http://multiwordnet.itc.it/english/people.php
3. Knowledge Management using Ontologies and the Semantic Web

Anottea [41] is a tool which provides the ability to share bookmarks among users, browser, or publish bookmarks on a server, thus allowing e.g. shared lists for collaboration. Interesting is the data model of the Annotea tool. It is also based on Jottings (or Bookmarks) and topics. Compared to the data model developed later the creation of keyword vectors is not supported, it has no property for signaling the visibility of a Jottings (public or private), and a description for the Jotting cannot be stored. An interesting extension is the introduction of the concept “Shortcut”. This allows to create metatemetadata, meaning that it is possible to create a Bookmark on a Bookmark, which is then called Shortcut. A prototype implementation of Annotea can be found in the experimental W3C web browser Amaya [1].

OntoShare [29] is a knowledge management system, which allows users to share documents using the Internet. OntoShare employs a user profile, which consists of topics from a shared topic hierarchy. A user can add a new document to the system and some annotation to this document. OntoShare than matches the document content with topics from the shared topic hierarchy. Other users are informed about the new document by the system via e-mail or when they log on to the system. Additionally a user can search for documents in the topic hierarchy. OntoShare also allows the evolution of an ontology. A user can assign a document to a different topic than suggested by the system. The system then tries to evolve the topic hierarchy by creating a new description of the topic using keywords and key phrases. A user can additionally suggest new topics, which are voted on by other users to be included into the topic hierarchy. The problems Davies et al. describe with the system can be summarized as:

- A tight integration with users work tasks was missing. To access the application was time consuming because the Java interface loaded too slowly in the web browser.

- Knowledge sharing as an altruistic act is problematic, which can be concluded from the fact that only a minority of users actually shared documents and a majority did not add any documents. Therefore the community was very dependent on the few people sharing knowledge.

- The user interface was problematic for new users and the ontology representation was not serving user needs well enough.

The last application we will take in consideration here is Bibster [26]. Bibster is a peer-to-peer system, which makes use of ontologies to allow the sharing of literature references. The system makes use of ontologies for importing data, formulating queries, routing queries, and processing answers. The system classifies new bibliographic entries according to two common ontologies. Queries can be formulated against the ontology, searching for an author, a special title, etc. Based on the expertise of a peer queries are routed to other peers. Answers are then returned to the peer querying. Duplicates are detected using semantic nearness. From own experience the author of this work is impressed by the functionality the tool provides, which seemed to work fine during tests. The major problem from a very superficial perspective seems to be the user interface, which makes the application hard to use in practice.

One can summarize the experiences from the described tools that they often do not fail because of technical problems, but because they lack a tight integration in users work tasks and do not provide an easy to use user interface. Often these tools
are only technology demonstrations, so that a good user interface is not necessarily on the agenda for the developers. To evaluate the usefulness of the tools in user studies though a good strategy for tight integration in the work environment is necessary as well as a well designed user interface.
As we have seen chapter 3 knowledge management tools often fail because of their lack of integration in users’ work tasks. Therefore an approach for the application designed here was needed avoiding this shortcoming. A methodology to ensure that a software supports users’ tasks and boosts the productivity of users is usability engineering. Usability Engineering starts with analyzing the work of potential users, thus providing a sound understanding of the work the future users are performing. With this understanding the users’ work processes are being redesigned supported by a new software application. Whilst the design process, ideas and concepts are continuously verified in collaboration with users. Refined prototypes of the application are presented from time to time to and commented by future users. This substantially rises the probability of delivering an usable and useful application.

This section provides an overview of Cooper’s Goal-Directed Design process, which is the basis for design process followed for creating the prototype described later. The description of Goal-Directed Design is taken form [27]. Afterwards a model explaining the search behavior of users is developed, which focuses on user goals. At the end of this chapter a survey of search engine user interfaces, which have influenced the design process, is provided.

4.1 The Goal-Directed Design Process

Cooper introduces for this process the term interaction design, it being “the definition and design of the behavior of artifacts, environments, and systems, as well as the formal elements that communicate that behavior.” [27] Following Reimann and Forlizzi Cooper describes interaction design as:

- “Defining the form of products as they relate to their behaviors and uses

- Anticipating how the use of products will affect human relationships and understanding

- Exploring the dialogue between products, people, and contexts (physical, cultural, historical)” [58]
According to Cooper interaction designers need to first and foremost understand the goals, motivations, and expectations (mental models)\(^1\) of the people whom they hope to design for. This can be done by using narratives, which have the important feature that they tell a story and therefore allow things to happen over a time period. Additionally these stories enable the interaction designer to develop empathy with the people, who they design for.

This leads to Cooper’s interaction design process called “Goal-Directed Design” as described in [27]. His process focuses primarily on the goals of the actual users of the application. Goals in the understanding of Cooper are addressed by functions and behavior and motivate the behavior of personas. These personas are archetypes of users, who represent a group of users with similar goals. Additionally he employs story telling, letting personas interact with the future software in scenarios. This enables the interaction designer to develop empathy with the people who he is working for. Further it is important in Goal-Directed Design to construct a clear picture of future users. Therefore Cooper suggests to employ techniques of ethnographic field study to gain mostly qualitative data on users’ goals, expectations, and applicable mental models.

Goal-Directed Design consists of five steps, such being (i) research, (ii) modeling, (iii) requirements definition, (iv) framework, and (v) refinement. A description of the process steps is given in the following.

### 4.1.1 Research and Contextual Inquiry

The research step focuses on understanding (future) users and the domain of the future application. Cooper suggests to use mostly qualitative research methods such as stakeholder interviews, subject matter expert interviews, user and customer interviews, user observation (ethnographic field studies), literature review, as well as product, prototype and competitive audits. Especially for the user observation Cooper mentions the methodology Contextual Inquiry Beyer and Holtzblatt introduce in their work Contextual Design [23]. Contextual Inquiry is conducted in the normal work environment of the interviewee, allowing the interviewer to understand the whole picture of the work environment. Cooper also talks about other types of research such as focus groups. For him especially focus groups are not a valid tool for research aiming to gather data for interaction design because people are rather unaware on how, when and what they do with products. Additionally focus groups tend to show only one view as the majority or loudest opinion is the one mainly perceived by observers, omitting necessary diversity.

### 4.1.2 Modeling Personas

In the modeling stage Cooper focuses mainly on creating archetypes of users, which he calls persona. Such a persona is carefully constructed form research data, but some fictional facts can be added for plausibility. The main point of personas is their representation of user goals. Personas summarize the behavioral variables of certain groups of people, such as their goals, attitudes, work or activity flow, skills and skill levels, and frustrations [27]. Important for personas is their ability to make members on the project team think of them as real people and feel empathy.

---

\(^1\)Mental models are a simplified explanation for a complex device or application. It is basically the strongly simplified explanation of how a “black box” works.
4. Usability

A persona as such tries to achieve three types of goals which need to be derived from research. (i) Life goals are the highest in the goals hierarchy and are long term goals of persons (e.g. achieving a promotion, be knowledgeable). (ii) Experience goals are below life goals. They present how people want to feel while using a product. (iii) End goals are the expectations of the outcome a user has when using a product (e.g. find an information, process an order, etc.).

Personas can have different functions such as primary personas, who are the main target for the design, secondary personas, who can be satisfied with additional features or negative personas, who are definitely not the target of the design.

4.1.3 Capturing Requirements with Context Scenarios and Needs

In the requirements definition step the interaction designer lets personas interact with the future application in narratives called Context Scenarios. These Context Scenarios tell a high-level story about how users will interact with the future application. A narrative is a powerful tool “[b]ecause interaction design is first and foremost the design of behavior that occurs over time, a narrative structure, combined with the support of minimal visualization tools such as the whiteboard, is perfectly suited for envisioning and representing interaction concepts.” [27]. The scenarios' content and context are derived directly from research. After developing the Context Scenarios the needs of personas are examined. These needs can be categorized as data needs (objects and information, which need representation in the system), functional needs (operations that need to be performed), contextual needs (relationships between objects and controls, which might need to be displayed together), and contextual requirements (such as the physical environment, as well as skills and capabilities of personas). Finally other requirements need to be taken into account such as business, technical, customer, and partner requirements.

4.1.4 Designing the Interaction Framework

The interaction framework consists of a robust high-level sketch of the application’s structure. It can consist of a paper-based prototype, which has some additional written descriptions explaining a user interaction with the application. Cooper explicitly mentions that a designer should apply interaction design principals, which provide guidance for designing the systems behavior in a variety of contexts, and interaction design pattern, which are template solution to common problems. The final step is to create key path scenarios, which explore the details that have been hinted in the Context Scenarios.

4.1.5 Refining the Form and Behavior

This step finalizes the design. If available and necessary industrial and visual designers are now helping to create a good look and feel. Then less traveled parts of the application are defined in validation scenarios. Then the design is finalized in a way, which allows a programmer to implement the design.
4. Usability

4.2 Modeling User Behavior for Searching

For an understanding of user behavior for the use case “search & browse” a review of available literature has been done. Due to time constrains a contextual inquiry could not be conducted deriving an understanding of how the Digital Library is used for searching and browsing. There are two communities which provide ideas on describing searches, namely the information science community\(^2\) and usability researchers. Neither of the communities came up yet with a model describing search behavior focusing on a classification of user goals. A brief overview over two models is given and then a model focusing on a classification of user goals is developed using ideas from the models described.

The topic of classifying searches is rather complex because it is dealing with human behavior. There are several ideas in the literature, how search behavior can be described, but the literature is unsorted and rather unspecific. This is surprising because searching is such an important issue for surfing on the Internet and a large market (see [17]).

One relatively detailed description of search behavior has been provided by Bates. In her articles Bates describes several ideas of how people search and what tactics they can apply. In [20] Bates identifies 29 tactics, whereas she defines tactic as a “move made to further a search.”. A description of those search tactics can be found in table A.1 in the appendix. The tactics are applicable to manual and online systems.

The major flaw of these tactics is that they cannot be used to classify a whole scenario from the perspective of goals. These tactics are rather little pieces that comprise a classification of a user’s search behavior. Therefore the tactics Bates describes are not sufficient to explain user behavior.

The tactics Bates introduces though can be supported especially by an application and user interface, which makes use of ontologies. In this sense another article of Bates is also relevant. In [19] she describes ways how a user can develop creative ideas to find a way of a search which has reached a dead-end (see in the appendix A.2). She describes 17 “idea tactics”, some of these being rather psychological. One example where an ontology-based application can support a user in searching are the tactics SUB and SUPER, which indicate the user should use a narrower or broader term.

Another approach to explain search goals and behavior is the work of Visciola. In his article “Search Types and Context of Use in the Semantic Web” [67] he identifies four different types of searches having a rising complexity. Thus being:

**Exhaustive** The search engine presents, almost everything that has been gathered on a given “subject” in the way the information has been gathered and stored, using its own criteria displaying the results.

**Question and answer** This search type has a fairly accurate query and a fairly accurate response.

**Explorative** The information is not accurately defined, the user is exploring a subject. All inaccurate answers needs to be reviewed to find an answer to the inaccurate query.

\(^2\)Information science here is the science of providing and organizing information. It is mostly concerned with digital and manual libraries.
4. Usability

Is there an answer? The user’s query will be unanswered as long as he does not reduce uncertainty and inaccuracy in his search. The information might be scattered in several documents or more generally gathered and stored in a way not satisfying the query. [67]

The work of Visciola shows some interesting aspects in defining search behavior. Nevertheless the categorization is hard to follow because it is not fitting the surrounding of a Digital Library very well. Visciolas work is focused on web search, which has some different implications, such as that metadata is not available in a well-structured way and the credibility of resources is difficult to determine. The author has adopted the work of Visciola summarizing his first two search types to “Finding a Specific Resource”. Additionally a more systematic description of his search types is developed, which allows the classification of search behavior using user goals.

Now the authors understanding is described, how a user’s search behavior can be classified. In a later section we will use this classification to develop context scenarios by letting personas interact with a search application.

- **Finding a Specific Resource** This resource can be an article or any other kind of information source which is digitally accessible. It is a search for a single target that is known to exist. For example the user wants to find a specific article written by a specific author, having a certain title. The termination criteria is reached when the article is found. The search criteria does not need to be as specific. A related case occurs when the user only remembers some keywords, but cannot pinpoint the article using e.g. the authors name or the title. In both cases the user knows that the article exists.

- **Finding an Unspecific Resource on a Specific Topic** This class of user behavior describes how a user is trying to find information on a field of knowledge he knows to exist. He has no specific target in mind though. This can also be described as exploring a specific topic. The termination criteria is reached when the information regarding a topic has been found e.g. an explanation of a certain technology, a tutorial for an application or technology etc.

  - **Finding an Unspecific Resource near a Specific Topic** This class is a specialization of the class “Finding an Unspecific Resource on a Specific Topic”. It implies that the user browses to a topic that is not the original search topic, but a related topic, finding useful information there.

- **Finding an Unspecific Resource on an Unspecific Topic** In this search a user is looking for an information of which he does not known if it exists. This information must be assembled from different sources to be useful and is an open ended search. Creativity is needed to reach a useful result. The termination criteria is vague, as there is not a clear target. The search needs to be split up into several subordinate searches.

This classification scheme allows to classify search behavior in regards to the different goals users have in mind. It cannot be claimed that a complete and sophisticated model to classify searches has been developed. But the author claims that this model describes search behavior at the level of detail needed for the later task of developing context scenarios.
4. Usability

4.3 Survey of Search Engine User Interfaces

This section focuses on describing other solutions for searching. One important observation on searching has been made by Quan et al. They identify in their work [57] that in a mechanical world the storage and retrieval systems are usually equal. If we want to store such things as a newspaper article in a file cabinet, we need to classify it for example as “public relations” and put the article into the appropriate file. Upon retrieval we can only find the article if we remember the correct classification. The only other method to find the article again would be to look through all files, which has prohibitive costs because it is very time consuming.

Even our computer file system works mostly that way. But in the digital world this is not necessarily true. If we are using links to files we already deviate form the basic principle that storage and retrieval system are the same. A computer system allows us several different ways to find a document again. We can classify the article not only as belonging to “public relations” but also as related to “Institute AIFB” and “SEKT project”. Hearst uses the same ideas in his work “Next Generation Web Search: Setting Our Sites” for classifying recipes [37].

Both the M.I.T Haystack [4] project and Microsofts “Stuff I’ve seen” [32] are aiming at the integration of all information source in one single access repository. In these access repositories they allow different ways of access to resources, such as sorting by date or people. The Haystack projects follows an approach using ontologies as underlying data structure. The Haystack front-end provides several orthogonal views on the data. In each view a user can narrow or broaden the scope of his search. The other views will then only show items that fulfill the criteria, which had been broadened or narrowed down.

Another interesting approach for a front-end in a digital library is the approach of Bier et al. [24]. In their work “Zoomable user interface for in-depth reading” they allow a user to browse through a collection of online articles, being able to zoom in and out of the collection, but keeping the fonts readable. Documents can be grouped semantically and their color indicates how much they have been read and how valuable they have been ranked by the user.

Important for this survey is of course the most successful (according to [42]) commercial search engine Google\(^3\). It is especially important to find out about the expectations a user will have. The most important observation regarding Google is that it is user friendly to connect search terms separated by whitespace with a boolean ‘and’. Another observation is the simplicity of the user interface.

Interesting is also the way how the social bookmark tool del.icio.us allows users to filter queries. The user interface of del.icio.us provides terms which have been used to classify bookmarks. The user can either choose to change the category by clicking on the word or can click on the word “add” displayed behind the keyword. The later action will create a new query connecting the added keyword with previous keywords using a boolean ‘and’. This allows the user to filter the results he has been viewing with the additional keyword. The result is that the intersection of the old category and the new category will be shown.

A scientific work regarding search engines is the already mentioned QuizRDF [30] (see 3.5). The tool uses a so-called “low threshold, high ceiling” approach, meaning that it allows normal keyword search, but if semantic annotation (e.g. with topics) is provided, it is exploited to the user. The query process consists of a

\(^3\)http://www.google.com
4. Usability

keyword search as entry point into the ontology. From there on the user can use the ontological concepts assigned to the resources retrieved by the keyword search as a starting point to browse along the provided ontology. Regarding the user interface, QuizRDF has a particular weakness in orientating the user. It is difficult for a user to answer the questions “where am I” and “where do I come from”.

The last interface, which plays an important role in this survey is the search interface of the BT Digital Library\(^4\). It shows no huge difference to other search engine interface, although some usability issues make it sometimes uneasy to use. For example is it problematic that the user has to choose manually, which resources (e.g. two available commercial data bases, books, subscribed journals, etc.) of the Digital Library he wants to search. An integration of search results is not provided.

\(^4\)In section 2.2 the Jottings functionality was introduced.
Part II

Design & Evaluation
Chapter 5

Proceeding Deriving Design

In this chapter the proceedings for deriving personas, scenarios, other requirements, the design of the interaction framework and finally of the prototype are described in chronological order. Good and bad experiences with the used processes are included and also problems, which came up during the design process. Finally some critical comments are made about the proceedings as outlined here. For example some activities, which were seen as essential for a process could not be followed. Therefore the here found approach is described arguing where results have to be interpreted with care.

5.1 Research

This work employs for many steps the Goal-Directed Design process as introduced by Cooper in [27] and briefly described in section 4.1. One major problem during carrying out the work on this thesis was the time constrained of being only for seven weeks at BT actually working on site and having future users available. Therefore it was impossible to conduct a contextual inquiry as suggested by Cooper. To understand the intentions and goals of users the decision was made in accordance with the project partner BT to analyze a previously conducted online survey once more. This allowed to start with the analysis step without being on site at BT. The quality of the analysis could not produce as in depth results as if a contextual inquiry had been done, but especially the free text answers of the survey allowed to see problems users had searching and using the BT Digital Library. The data analysis done by BT was reviewed also and used as a data source (see [14]). Additionally experiences from tools such as OntoShare [29] or Bibster [26] are taken into consideration. They especially suggested that a design was needed, which is very much integrated in the normal work tasks of users, making the seller role in knowledge sharing (see 2.1) easy to fill. An important finding was that the application should be of personal value, to the person sharing information. The value of the described applications was derived by a small group of users willing to undertake altruistic actions on behalf of the rest of the knowledge community.
5.2 Deriving Personas and Context Scenarios

After the research step the Goal-Directed Design process could be followed more closely. From the data of the research step personas and context scenarios were derived. For the two use cases “Search & Browse” and “Jottings” were two personas identified. Daniel is the primary persona for the use cases, meaning that he is the one the design effort is focused on. Joe is the second persona derived from research data, being a secondary persona meaning that the design includes some extensions, thus fulfilling his needs. Generally personas allow a designer to see the world through the eyes of a user and feel empathy with the future users. Especially the goals of personas are important to allow a smooth work flow throughout different tasks. The personas have been reviewed by the head of the Digital Library and two other persons working on the SEKT project at BT. Some rather cosmetic changes were made after the review.

In the next step context scenarios were derived, in which personas interacted with the future application. For the use case “Search & Browse” three scenarios were written, namely “Finding a Specific Resource”, “Finding an Unspecific Resources on a Specific Topic”, and “Finding an Unspecific Resource on an Unspecific Topic”. The different scenarios match the classification of search behavior as introduced in chapter 4.2. The first scenario “Finding a Specific Resource” has two settings, which reflect the fact that a user can remember very precise facts about a document such as title or author, but also rather weak facts, maybe a word describing the context of the document. A similar situation can occur in the scenarios “Finding an Unspecific Resources on a Specific Topic”. Therefore a specialization is introduced called “Finding an Unspecific Resource near a Specific Topic”, where a resource is found by navigating away from the actual topic to a different topic. This specialization is very similar regarding the user’s search behavior and therefore the scenario already includes the specialization. The classifications of user behavior have been reviewed also by the head of the BT Digital Library and two members of the SEKT project staff. After the review the specialization of the classification “Finding an Unspecific Resources on a Specific Topic” namely “Finding an Unspecific Resource near a Specific Topic” was added.

The scenarios on “Jottings” are straight forward. They explain how a user can add a Jotting to the application, edit, delete or make use of a Jotting. All scenarios mentioned up to here have as persona Daniel, indicating that they envision a user who does not have expert IT skill. Also all scenarios mentioned up to here serve as input for the design of the interaction framework and the prototype.

The scenarios on “Advanced Ontology Adaption” could not be taken to a further stage in the design process due to time constrains. These scenarios would have been very complex to design because they introduce some rather complex tasks such as the scenario “First Installation of Knowledge Sharing Application”. They would have most likely required also a more sophisticated visualization technique than DHTML, which would have made it harder to create a design. These scenarios are included in this work as a means to provide ideas for a more sophisticated knowledge sharing application. The scenarios were evaluated with three members of the SEKT project. With two members of the SEKT project the scenarios on “Search & Browse” and “Jottings” were reviewed in a meeting, which also served for creating ideas for the design of the interaction framework. In this meeting a rough data model was proposed, which is described in section 7.3.

Personas have certain needs which have to be satisfied by a user interface. As
mentioned in section 4.1.3 these needs can be categorized as data needs, functional needs, contextual requirements, and contextual needs and requirements. The needs for the two use cases are not explicitly listed, but kept in mind while designing the interaction framework as well as the prototype. This is also due to the time constraints in finishing the prototype design to be able to evaluate the design. The same applies to other requirements. These are requirements in a more traditional system design view, which can be found in a functional specification document. Here only some issues that came up during the design process are mentioned.

5.3 Design of Prototype and Evaluation

The design of the prototype began by sketching a paper & pencil prototype of the screens, which are important for the interaction with the system. After the design of a relatively stable paper & pencil prototype the first evaluation was done with four users. All of them were members of the SEKT project team though they were all not directly involved in the design and implementation of the application. The paper & pencil prototype was shown to the users with some explanatory comments on how the functions of the prototype work. The test persons were asked to comment on issues they liked or did not like. This evaluation was done in an informal way. Results were recorded by the interviewer in written form and not analyzed formally. The comments of the users were of course regarded in the further design of the prototype.

For the design of the prototype Microsoft PowerPoint was used as a tool to display pictures on a computer screen. With a graphics program the screens of the application were constructed in a highly believable fashion. This means that the design of the application was placed in an empty browser window. The use of a graphics program was more flexible and less overhead then actually coding a prototype of the user interface in HTML. The screens were put together in a story board, showing how a user could navigate through the application. Then followed another evaluation of the prototype. This evaluation was carried out more formally, by selecting users, who were not involved in the SEKT project. The interviews were conducted in a more formal setting. The formal analysis of the results is described in chapter 8. After the interviews the prototype was revised to remove issues users had pointed at. A third evaluation could not be carried out due to time constrains and to the fact that the author had no access to possible users as he was not on site anymore.

5.4 Remarks on the Proceeding

Finally some remarks need to be made. A problem was that at the beginning of the design only a rather coarse description of the DILIGENCE process existed because the process is still being developed. It was very unclear up to the finalization what was exactly meant by the requirement that users should provide reasoning why they adapt the ontology. In the description of the process at hand now ([66]) this is better defined. A careful consideration might allow to actually construct prototype reasons described later in the scenarios. The rather coarse grained description did not allow this.

1In this case some more awkward ways were also included to show the whole functionality of the prototype.
Also the before mentioned necessity to omit the contextual inquiry, which would have allowed the author to create his own picture of the work of the knowledge workers was a problem. It is hard to actually determine how well the personas are constructed. Also it would have been interesting to observe user behavior on search in a contextual inquiry and then start from there, trying to develop a model explaining the behavior.
Chapter 6

Requirements Analysis: the BT Digital Library

In this chapter the results of the requirements analysis for the new applications for the BT Digital Library are described. The requirements analysis is generally focused on providing a stringent interaction design, but additional requirements are also captured if needed to create a sound understanding of the application domain. This sound understanding of the application domain is a prerequisite for a good interaction design.

Two use cases defined for the BT Digital Library case study as part of the SEKT project are examined here, namely “Search & Browse” and “Jottings”. The use case “Search & Browse” concentrates on exploring the possibilities of how users search through an ontology-based Digital Library. The design challenge is here to propose a user interface, which allows users to find more relevant things more easily.

The use case “Jotting” looks into a knowledge sharing situation, where users share so-called bookmarks or favorites\(^1\). The challenges for the use case “Jottings” are the tight integration into the work tasks of users and providing an environment, where the sharing of documents is not an altruistic act, but an additional value to a task that a user would perform for his own use.

This chapter visits the BT digital library, then general requirements are described. Afterwards the two derived personas “Daniel” and “Joe” are described, the scenarios for the use cases “Search & Browse” and “Jottings” follow. Then scenarios on “Advanced Ontology Adaption” are illustrated, which are mainly included for envisioning the way for an extension of the application allowing more sophisticated changes of the ontology. The scenarios describe their intention, the expectations of the personas interacting with the application, the setting and then the description of the actions a persona undertakes. Finally a summary and a conclusion of the requirements analysis is provided.

\(^1\)Bookmarks and favorites are the combination of a URL and a brief description of the document’s content the URL points at.
6.1 Initial Implementation of the BT Digital Library

To give the reader a short introduction what the digital library at BT is and its purpose [14] is cited:

“BT began building its Digital Library in 1994 and over ten years has developed an online system that offers its users personalization, linking to full text from abstracts, annotations tools, alerts for new content, and the foundations of profiling. A key driver in developing the library has been the desire to provide a single interface to the whole collection, drawing together content from a wide variety of publishers. As one user said in a recent survey, the BT Library is “often better than the university library because you can search all resources and get a summary, perfect for research / initial project work.”

The BT Library allows its users to search the library’s contents. In addition, they can browse through “information spaces” that have been created on topics known to be of interest for people in the company or through the contents of journals in the Library. Information Spaces bring together content form the library’s databases and details of new books into a single page in the Library. People can “join” an information space to be alerted to new articles on the topics and can create their own private information spaces for topics of particular interest to them.

The Library contains abstracts of all relevant technical papers, and the full text of more than a third of all the relevant engineering literature - a million articles form over 12,000 publications, including journals, conference proceedings, and IEEE Standards. This is provided in the form of two databases, Inspec and ABI / Inform. As well as technical literature, the Library holds almost 1,000 management and business magazines on-line, allowing people to track developments in management thinking or corporate development over more than ten years. The library also provides on-line access to a number of the O’Reilly technical books. In addition, the library makes recommendations to readers about books which they may wish to purchase, based on their membership of information spaces.

The Library uses software developed by BT’s Next Generation Web Research to power its searching and browsing.”

6.2 General Requirements

In this section requirements on the application are described, which are not covered by the context scenarios. These are mainly technical or business requirements. The requirements list presented here is not claimed to be complete, as mostly requirements are listed, which have an effect on the design of the prototype.

Research requirements:

- Support of the DILIGENT process This requirement was made as one important goal of this work to explore how an application supporting DILIGENT
would look like. This goal was set by the Institute AIFB from the University of Karlsruhe.

- **Explore the use of ontologies for searching and browsing**: Exploring the use of ontologies for searching and browsing electronic content is a goal of the SEKT project. The application design needs to make use of ontologies for searching and browsing as a show case.

**User requirements:**

- **Knowledge sharing not as an altruistic act**: Experience with other knowledge sharing applications such as OntoShare at BT has shown that it would be helpful to create knowledge sharing applications where the seller gains a benefit himself from adding content, which is then distributed to others.

- **Privacy**: During the analysis of the online questionnaire, which was also used for the BT Digital Library Initial Analysis [14], the issue of privacy was raised. It was important for users that their behavior is not constantly observed and they need to be able stay anonymous if they want to. This also suggested the requirement to let users decide to keep Jottings private, thus not sharing them with other users.

- **Users need to stay in control**: Users want to stay in control of the system. They generally distrust the ability of the system to remove them for example from alert list. The implication is that users should be allowed to control what the system does on their behalf.

- **Sharing knowledge with people outside BT**: From the analysis of the online questionnaire also the requirement came that it is important for some users to share Jottings with people who are not BT employees. This case can occur when dealing with customers or partners.

- **Push and pull approach distributing knowledge**: From good experience with both kinds of approaches at BT the requirement was made that there should be a push as well as pull approach to knowledge sharing by the application. A push functionality would be e.g. an e-mail or an instant message send to a user advertising new Jottings. A pull approach is the possibility for a user to request knowledge, such as browsing for a certain Jotting or the BT Digital Library.

- **Search history**: A history of the search path should be provided to allow users to trace back their search, thus making it easier to find useful resources again.

- **Results integration**: Right now the BT Digital Library uses several data sources a user has to manually select from for a search. These sources should be integrated to enhance the usability as users would not have to think about searching a particular data source.

**Technical requirements:**

- **User interface accessible via a webbrowser**: This requirement was made because the BT Digital Library as of now is accessible via the BT Intranet. Therefore the technology of choice is to use DHTML allowing an easy integration into the existing infrastructure.
6. Requirements Analysis: the BT Digital Library

- **Duplicates** Because in both use cases the application is dealing with a number of data sources a capability to detect duplicates in a result set must be implemented, thus displaying doublets as one resource. It may be indicated by the application if a resource can be obtained from several sources.

- **Ontology alignment** Right now two thesauri exist, one for the ABI/INFORM database and one for the INSEPC database. These thesauri (or ontologies) need to be aligned or merged. This would also make the detection of doublets easier because they would be semantically very near in the ontology.

- **Ontology refinement** It should be considered to refine the ontology (e.g. using ontology learning) in a way that a topic being at the end of a subTopic relation has about 50 instances or less. This would allow a user to explore a topic fully because not an overwhelming number of results are assigned to one topic.

- **Searches making use of particular metadata** The search functionality must make use of different metadata such as publication date, publication type, and author. If more types are found they should be included. One kind of metadata are topics. A topic in this case is a descriptor for a well-defined theme or subject. It can have relations to other topics such as being a subtopic (hierarchical link) or be a related topic. Topics are unambiguously defined and their common understanding by users of an ontology needs to be assured.

6.3 Personas

In this section the personas Daniel and Joe are described. Personas are prototype users representing a user group of an application to be designed. Personas are derived from research data, but some gaps and personality descriptions are filled with fictional facts to make the persona more believable. This allows a designer to see the world through the eyes of these personas rising the odds of designing a useful application substantially (see on personas also 4.1.2). The persona Daniel is the primary persona for both use cases, meaning he is the one the design is made for. Joe is a secondary persona, being satisfied with the user interface for Daniel, but has some additional needs. Basically the persona Daniel is a normal office worker with average level of IT skills and Joe is a persona with high IT skills.

6.3.1 Persona Daniel - A Typical Library User

Daniel is 27 years old and has been working for BT since 3 years. He joined BT after he graduated and holds a marketing degree. His IT skills are basic, enabling him to manage MS Office tools and web application well. He usually stays with the basic features trying not to spend a lot of time in learning a new application. Daniel has insight on larger industrial companies and works for BT wholesale in the health sector. He sometimes works in the office, but at other times he his traveling to visit customers. He uses frequently his desktop machine at the office and a laptop when he is at a client site.

Sometimes he talks with his colleagues about new developments in the industry, but is rather reluctant to share to much of his knowledge with them in fear to loose competitive advantage. But he usually tries to be helpful if a colleague asks him directly for help. He dislikes having to search a lot to find a particular piece of
information. He gives up rather quickly because he is usually very busy. He likes if information is pushed to him in a right quantity rather than having to look for it himself. He also dislikes the idea that his e-mail communication might be monitored by an application, as he has privacy concerns.

He logs on to the BT Digital Library roughly once week as he has to do some assignments for his MBA. Usually he spends about one hour searching for information and about one hour analyzing the found information per week. He mostly uses the Library for his MBA assignments. The business journals are a good help for that. Sometimes he also uses the BT Digital Library to write business proposals as he needs to research what technology his client wants or what kind of business model might be appropriate for a certain kind of client. He also likes to read background information on the telecommunications industry. He likes the way the BT Digital Library gives him easy, inexpensive access to papers on business topics. He is also particularly interested in up-to-date information on issues he is dealing with in his job. As he is less professional in typing, he likes the spell checker in the Google search engine.

Daniel’s way of organizing knowledge assets on his computer is rather simple. He has a flat structure of folders and he rarely uses bookmarks. He would be thankful if there would be a tool available that could help him in organizing his knowledge assets. He would rather search again than saving the information somewhere. This means he is loosing time to find information again. He also rarely uses advanced functions in search engines as he does not know how his search is influenced by different parameters.

6.3.2 Persona Joe - An Expert User

Joe is 45 years old and a domain expert in grid research. He has been with BT for 21 years. He works at BT exact and is doing research on internet-based computing models. He quickly learns new applications. He makes use of advanced features if it helps him to achieve his goals. In general he is working at his office at Adastral Park, where he uses a desktop machine. Sometimes he works from home using his computer at home to log on to the BT intranet because there he is not being disturbed by others. He joins his colleagues for lunch or tea breaks. Sometimes he talks with them about their on going projects and they share information.

Joe likes to help others with their problems when they give him the chance to decide about when he can help them. He uses the search engine Google a lot and it is his personal benchmark to judge about other search engines. As it is critical to his research he likes the quality of the information available at the BT Digital Library. Joe dislikes the small font used in some applications as he has sometimes problems reading it. He is opposed to his e-mail communication being monitored even if that would make his knowledge organization easier.

As part of his work is doing research, he uses the Digital Library about once or twice a week. He is in favor of the high quality information available there. He spends about 4 hours searching for information each week and also about 4 hours analyzing the found information from all kinds of sources including the Digital Library. When he is using the library he usually needs some information for his research. Sometimes he also uses the BT Digital Library to solve a technical problem he has, or which one of his colleagues asked him about. Sometimes he also wants to read a bit about new emerging technologies, which might effect his job or he has personal interest in.
He tries to organize carefully knowledge assets that are on his computer. For his files and for his bookmarks he uses a relatively deep folder structure. Due to the large amount of bookmarks and due to the fact that knowledge structures evolve over time he is a bit frustrated by the approach. In spite of the shortcomings he tries to stay with his careful organization because of the lack of an alternative.

6.4 Context Scenarios on “Search and Browse”

Context scenarios illustrating how a user interacts with the new user interface searching and browsing through the BT Digital Library. The classification of user search behavior as developed in section 4.2 is used to construct the different scenarios. The interface is constructed for the persona Daniel. Joe as secondary persona is completely satisfied with the application as it is described here, except for the usage of a special syntax in the search field. This syntax is described later in the prototype design (chapter 7) and allows the search for example for a certain topic or author from the text field.

6.4.1 Scenario “Finding a Specific Resource”

Intention of Scenario: A resource in this case can be an article or any other kind of information source which is digitally accessible. It is a search for a single target that is known to exist. For example the user wants to find a specific article written by a specific author, having a certain title (setting 1). The termination criteria is reached when the article is found.

The search criteria does not need to be as specific. Another case, described in setting 2 is when the user only remembers some keywords, but cannot pinpoint the article using e.g. the authors name or the title. In both cases the user knows that the article exists.

Expectations: Daniel wants to be able to find things as fast as possible, with the least effort possible. He does not want to fill out many text fields. If he knows the title of a resource he will use that, if not he will use keywords he remembers and enter with a white space in between each keyword. He expects that these keywords are connected with an boolean ‘and’ comparable to the search engine Google. If he has entered the title of a resource he expects to find it at the top of a results list. If he used keywords he expects the resource to be somewhere on the first results page.

Setting 1: Daniel needs to find a paper for an assignment for his MBA class. His paper is on how digital media changes the broadcasting sector. He knows that the title of one article to start with is “The convergence of telecoms and broadcasting services” because his professor gave it to him.

1. Daniel opens his web browser and navigates to the BT Digital Library.
2. The application provides a text field for simple searches. He enters the title of the paper “The convergence of telecoms and broadcasting services” and starts the search.
3. The search engine displays instantly results that assumably match his search target. The application provides also options to refine his search.
4. Daniel finds the paper he is looking for to be the first item in the results list.

5. He views the paper and prints it because he wants to read it at home.

**Setting 2:** Daniel remembers that he has read a paper about the topic he is writing about a couple of days ago. He does not remember the title nor the author, but has a clear idea of the topic and what the author talked about. This means Daniel will enter keywords that describe what the author talked about.

1. Daniel opens his web browser and navigates to the BT Digital Library. The initial search screen is shown. As Daniel does not know the exact title nor the author, he remembers that the article was about “webcasting” and “network”.

2. He enters the keywords “webcasting” and “network” and starts the search.

3. A result list is displayed in his web browser and additionally he is given the possibility to refine his search by using topics from the topic ontology. As Daniel entered an ambiguous term the result set can be changed to display only results regarding to a certain meaning. Here the term network is disambiguated into “computer network” and “broadcasting network”. Daniel chooses “computer network”. The result set changes and displays only results that regard computer networks.

4. Daniel does not find the paper he is looking for in the first ten results of his search. He uses the other features provided on the page to refine his search and excludes papers written before the year 2002. Now he finds the search target within the first ten results and views the paper “Glowpoint branching out into Webcasting”.

5. He prints this paper to read it at home.

6. He goes to the printer and picks up the printed pages.

### 6.4.2 Scenario “Finding an Unspecific Resource on a Specific Topic”

**Intention of scenario:** This scenario describes how a user tries to find information on a domain of interest of which he knows it exists, but has not a specific target in mind. This can also be described as exploring a specific topic. The termination criteria is reached when the information regarding a topic has been found e.g. an explanation of a certain technology, a tutorial for an application or technology etc.

**Finding an Unspecific Resource near a Specific Topic** This is a specialization of the scenario “Finding an Unspecific Resource on a Specific Topic”. It implies that the user browses to a topic that is not the original search topic, but a related topic, finding useful information there.

**Expectations:** Daniel has basically the same expectations as in the scenario “finding a specific resource”. The only thing that has changed is that he is a bit more patient. He will look at the first result page closely and maybe the second one. If he is still not satisfied he might search for a second time with different topics.
Setting: Daniel prepares to write a business proposal for an important client. The client wants to start a new business branch in webcasting. The client explained Daniel his business model “On-Demand Webcasting”, but Daniel does not know enough to judge about the potential. Therefore he decides to search the BT Digital Library for related ideas. Daniel hopes that this will give him an additional insight so he can write a more customer focused business proposal.

1. Daniel enters the name of the technology “On-Demand webcasting” the client mentioned.

2. The application presents Daniel with a set of results. Daniel scans some of the papers, one giving a pretty good insight.

3. Daniel changes the search topic by clicking on one of the related topics that are displayed. He clicks on “Intranets”. Daniel scans through the papers displayed. But likes none. He goes back to the first topic and selects another related topic, namely “streaming media”. Here he finds the article “A do-it-yourself Webcasting kit” that really interests him because it gives a different perspective on On-Demand Webcasting.

6.4.3 Scenario “Finding an Unspecific Resource on an Unspecific Topic”

Intention of Scenario: In this search a user is looking for an information of which he does not known if it exists. This information must be assembled from different sources to be useful and is an open ended search. Creativity is needed to reach a useful result. The termination criteria is vague, as there is not a clear target. The search needs to be split up into several subordinate searches.

Expectations: Daniel is only occasionally searching on unspecific topics. He feels that he does not have a lot of time to spend on things that are not directly related to his work. Usually he reads only about new developments when information is push to him by others. At rare occasions, when he wants to be ahead of colleagues or when a client suggests business models or technologies he does not know he will search on his own. Daniel knows that this will take a bit of his time, but he will rarely spend more than 15 minutes on a search at once.

Setting: Daniel has to write another paper for his MBA. His goal is to give a market overview over a market niche that might also be interesting for BT. Therefore Daniel is doing an iterative search, splitting his search into several different topics. He wants to find out what competitors are doing. He looks in the BT Digital Library to find information in some business magazines about the competitors and their business models. He will additionally use the web to look at competitors web pages.

1. Daniel enters the name of the market niche.

2. He finds two or three articles that mention the market niche. Daniel is not really satisfied with the result as the information is too unspecific for him.

3. The application shows him synonyms of the words he has entered. This helps Daniels search because the market is still evolving and no clear name is established yet. Daniel chooses to include also the synonyms in his search.
4. Now Daniel finds six articles that are interesting. He saves those on his hard drive. He will read them later. Now he wants to continue his search.

5. Daniel enters the name of a competitor. The application returns a set of articles mentioning that company. But Daniel is not satisfied with this result because he is particularly interested in the companies engagement in the market niche he is investigating.

6. Daniel adds from a list of recently used topics, the topic which represents the market niche and then starts the search again.

7. The application returns now articles which are only concerned with the market niche and the company he is investigating. Now Daniel reads the section in the articles returned that he is interested in.

8. He copies the important parts of the article to a Microsoft Word document to have the essential information in one place. He also copies the complete reference to the original document in this Word document to be able to find the source again.

9. Daniel does the same search on a few other companies.

10. Daniel saves all the topic and keywords he used for the search and the result sets as a view in his knowledge management system.

11. Daniel starts to write his paper with the information he has retrieved so far.

12. He finds out that he misses some information that he did not consider within his first search. Daniel had already closed the application, but he is able to load the view he had stored in his knowledge management systems again to continue his search.

13. The application loads the search (including used topics, and their combination, as well as the old result sets). Daniel starts to look for the details he is missing. He browses a bit in the old result sets. But he cannot find the desired information.

14. Daniel starts a new search by using some of the topics he has used before, but additionally using some new topics.

15. The application returns a new result set that it will also stores together with the old results and topic combinations. Daniel browses the first few results and sees that he actually found what he was looking for. He continues to write his paper.

16. Daniel stops to work for today and goes home.

17. The next day Daniel is again back in his office. During a longer lunch break he continues to write his paper. He starts Microsoft Word, opens the document he had used and looks through the parts he has already written. Now he finds that he stopped yesterday at a point where he was missing an information. He loads the search view again and looks for the missing information again using some of the topics he had used before.

18. Daniel finishes a draft version of the paper and therefore also his search.
Additional remarks A further development could be to enhance the browsing with concepts such as zoomable interfaces [24]. A zoomable interface allows a user to gather a collection of documents for studying. The documents are sorted in clusters and their color indicates the personal rating how often the document had been read. The user is able to read the documents by zooming in and out of the clusters of documents. The application ensures that text of the document is readable in different magnifications. Additionally it should be very easy to extract references from the available metadata, thus allowing to create easily a bibliography.

6.5 Context Scenarios on “Jottings”

The interaction with the application for the scenarios in this section is straightforward. It is derived from using http://del.icio.us [2] and the BT Digital Library Jotting function. Additionally users are enabled to assign topics to the URL of a Jotting they are storing. The issue of sharing bookmarks with external partners is also addressed. Here a topic has a well defined meaning as opposed to a key word, which can ambiguous. The scenarios described here have Daniel as primary persona. Extensions for advanced users like the persona Joe are described in section 6.6.

6.5.1 Scenario “Add a Jotting”

Intention of scenario: The user adds a new Jotting. The application provides the user with as much derived information as possible, including the URL, title, possible topics and if available a copy of another user’s description of the Jotting.

Expectations: Daniel expects that he will be able to store links to web pages for future retrieval easily and that technology will help him to structure his links in a way that makes it easy for him to retrieve those links again. Creating new Jottings needs to be integrated in his work process tightly because Daniel is not willing to spend a lot of time on it.

Setting 1: Daniel is browsing on the Internet and finds by coincidence a page that a client should find useful for convincing his top management that they should invest in a certain kind of technology. The article describes how one of the client’s competitors already uses the technology to give them a competitive advantage. Additionally, Daniel wants to share this information with some of his colleagues.

1. Daniel clicks on a button in his browser’s Jotting toolbar to create a new Jotting.

2. A new window opens, which includes the URL of the page, the title, suggestions for topics to assign to the web page, and a blank text field for a personal description. All of this information can be edited. Additionally a text field for entering e-mail addresses is shown. Entering an e-mail address will push the interesting link to other users.

3. Daniel is satisfied with all of the descriptions, but he wants to share the information with others. He enters the names of the people he wants to notify including the client. The application will access his address book and insert in the complete names and their e-mail addresses.
4. The application generates an e-mail that includes the classification and Daniels description. The e-mail will eventually be sent to the e-mail addresses Daniel had entered. The Jottings will from now on show up in the personal topic ontologies of Daniels colleagues with a description indicating that Daniel has thought this Jotting was useful to them.

5. The delivery of the e-mail is delayed by the application. Each user can choose how often the e-mail will be delivered. Daniel did not change the default option, so the notification e-mails are delivered once a day to him.

**Setting 2:** Daniel has found a useful web page explaining new technologies in a way that is suitable for his technical background. He decides to store the link.

1. Daniel clicks the “Add Jotting” button in his browser’s Jotting toolbar.

2. A new window opens that already including the URL of the page, the title, suggestions for topics to assign to the web page, and a blank text field for a personal description. All of this information can be edited.

3. Daniel looks at the topics and recognizes that one of the suggested topics does not match the content of the page.

4. Daniel communicates to the application that he wants to change this topic.

5. The application displays a window showing the topics related to the topic that Daniel wants to change. The application also allows to search for a topic which is not semantically near to the original topic. Daniel finds an appropriate topic semantically near to the original topic and selects it.

6. Daniel is still not satisfied and believes a topic is missing. Therefore Daniel wants search for the missing topic and if he cannot find it he wants to add it.

7. Now the application shows again a text field to search for a topic. Additionally semantically related topics with the ones chosen, are shown. This enables Daniel to start browsing the topic ontology.

8. Daniel chooses to start with the search function and enters keywords naming the topic in the correct text field. The keywords Daniel entered do not match any of the topics available in the ontology, but the application has found some topics which are semantically near to the entered keywords and displays those.

9. Daniel chooses one of the retrieved topics, therefore he does not need to add a new topic to the ontology.

10. Daniel confirms that the Jotting is now correctly classified.

**Setting 3:** Daniel has found an interesting web page that is connected to the project he is currently working on.

1. Daniel currently views this page. Daniel clicks on the button “Add Jotting” in his web browser.

2. The system processes the web page and suggests topics for Daniel to assign to the Jotting.
6. Requirements Analysis: the BT Digital Library

3. Daniel is not happy with the suggested topics and communicates to the system he wants to change them.

4. The application provides Daniel with a user interface that enables him to browse the topic ontology to find an appropriate topic. Daniel browses through the topic ontology, but he does not find an appropriate topic.

5. Daniel decides that he wants to extend the topic hierarchy to his needs. Daniel chooses to create a new topic.

6. The application prompts Daniel to provide a name for the new topic and optionally a short description. The application asks Daniel now why he introduced the topic. The application provides Daniel with a number of prototype reasons, but he can also choose not to give a reason as well as provide a his own reason in a text box.

7. The application now prompts Daniel to confirm that he would like to classify the Jotting with the new topic.

8. Daniel confirms this and the Jotting is classified against the new topic.

Additional remarks  At the moment only a simple way of extending the ontology is introduced with the application. Any other more sophisticated solution for extending the ontology is difficult to realize with DHTML technology because it would require advanced visualization and interaction techniques. A possible solution using Internet technologies might be to use Macromedia Flash, which can deal with XML and is by its nature a vector graphic format. Therefore it could be used to visualize an ontology. Macromedia Flash also has the capabilities providing richer support for interactions. Another solution would be to use a Java-applet, which would also allow the usage of (existing) Java libraries. All these solutions have the disadvantage that the Java applet or Maromedia Flash movie needs to be loaded, which could take too long for a reasonable integration of the application into users’ work tasks.

6.5.2 Scenario “Use a Jotting”

Intention of Scenario: This scenario describes how a Jotting can be used to recall web pages. This functionality helps the users to achieve their goal of finding information again, which they had viewed before. Users are given a personal reason to enter information. Therefore knowledge sharing is a by product rather than an altruistic act. The knowledge seller as described in section 2.1 can makes use of his own actions.

Expectations: Daniel expects that he will be able to find webpages faster and more easily than he used to do. Daniel also expects to find useful Jottings others had previously stored.

Setting: Daniel wants to revisit a web page, which he had created a Jotting for a couple of weeks ago. The page described a competitor’s business model. He does not remember the correct name nor the exact topics that he used to describe the web page.
1. Daniel opens up his web browser. Daniel thinks of the names of the concepts which would best fit the classification, and enters those in a text field used to search for Jottings. He chooses the option to search only in his own Jottings.

2. The results are shown on a generated web page. One of the keywords Daniel entered did not match any topics provided by the topic hierarchy. So the application mapped it to an appropriate topic. Additionally, related concepts are shown, so that Daniel is able to browse through the ontology to find the resource he is looking for. Many of the topics from the shared topic ontology are hidden because they have not been used to classify any Jottings. Some higher level concepts are shown for navigation purposes.

3. Daniel looks at the results presented by the application. He does not find his search target, and therefore he clicks on one of the topics that are displayed to be related to the keywords he had entered and starts browsing the topic ontology.

4. The application retrieves the Jottings classified by the topic Daniel had clicked on.

5. Daniel finds the correct link, clicks on it and the correct web page is displayed.

6.5.3 Scenario “Edit a Jotting”

**Intentions of Scenario:** The user has the option to edit the URL and the other assigned metadata because the Jotting might have been inaccurate. The URL or the content might have changed, as well as the metadata may not be appropriate anymore.

**Expectations:** Daniel might want to edit a Jotting, if a description is incorrect or the page has moved and he knows the new location. Again the process needs to be integrated tightly with Daniels work procedures.

**Setting:** Daniel is browsing through his Jottings to find a page that he knew he had entered a couple of months ago.

1. Daniel finds the link and clicks on it. The browser displays a 404 error message including a referrer. The website changes to the referred web page.

2. The application asks Daniel, in an unobtrusive way, if the new URL is still valid, together with the metadata (topics and description) of the Jotting.

3. Daniel looks at the page and finds out that this is the case.

4. Daniel wants to change one of the topics that have been used to classify the web page. He marks the topic and communicates to the application he wants to change the classification. The application suggests some topics semantically near to the old topic.

5. Daniel has a different topic in mind. He searches for the topic by entering the topic’s name in the appropriate text field. Although none of the available topics match Daniels request, he finds an appropriate topic, which he chooses to use.
6. Daniel communicates to the application that he has finished editing the Jotting.

**Additional remarks**  The application will access the web pages of all Jottings in certain time intervals. If a 404 error message is received that Jotting will be marked having a broken link.

### 6.5.4 Scenario “Remove a Jotting”

**Intention of Scenario:** The scenario allows the user to remove the metadata and the URL form the user’s Jotting perspective. The Jotting will be marked as being deleted by a user. Although other users will still be able to see the Jotting, they will be informed that the Jotting has been deleted by another person.

**Expectations:** Daniel expects that the Jottings are always up-to-date and that there are no broken links. Occasionally he finds that a web page is out of date, so he wants to remove it from his Jottings to ensure he does not use information that is out of date.

**Setting:** Daniel uses one of his Jottings to open a web page. When he looks through the content of the web page he finds that the information has not been updated for a while and might be misleading. Daniel decides to delete the Jotting.

1. Daniel clicks on the “delete Jotting” button in his browser. The button is only activated if the URL has been recognized as belonging to a Jotting.
2. The application will no longer display the deleted Jotting anymore to Daniel.
3. Other users who have also jotted the same link are being shown that a user has removed the link, which indicates that it might be out of date. The indication might look like: (Removed by 1 user).

### 6.6 Context Scenarios on “Advanced Ontology Adaption”

This set of scenarios describes how users who are untrained in ontology engineering can be enabled to extend an ontology. The DILIGENT process is used for an understanding of how changes to the ontology are introduced. Users not trained in ontology engineering will be provided with an initial ontology created by domain experts and ontology engineers. The user will be able to extend the ontology as described in the “local adaption” step of the DILIGENT process.

**Providing reasons for ontology adaption**  The DILIGENT process suggests that users, who are extending an ontology should provide reasons for making the changes. It is likely that users will be reluctant to do this because it is an extra task which is not necessary in their perspective. One possible solution is to provide a number of prototype reasons that a user could select from. Additionally the user should have the opportunity to give an other reason than the suggested prototype reasons or refuse to give a reason.
6.6.1 Scenario “First Installation of Knowledge Sharing Application”

Intention of Scenario: This scenario does not necessarily link very well with the other Jottings scenarios. It applies only if a local knowledge base is to be installed or a centralized solution should import knowledge that is stored locally. A knowledge base should be initially populated with resources already available to avoid the so-called “cold start” problem (meaning that the application is not of use without a lot of entered data). Populating the knowledge base with resources may be a labor-intensive task that can only partly be supported by natural language processing and ontology learning (compare [66]). This process can, but does not have to be omitted for an application that is using Jottings. The bookmark files of a user could be accessed, the stored links could be retrieved, analyzed and classified.

Expectations: Daniel expects that the installation process will run smoothly without requiring him to provide a lot of input. He installs the software with the expectation that in the future his bookmarks will be better organized. Daniel expects that the installation process will take approximately 10 minutes.

Setting: Daniel has heard from a colleague that BT has developed a new application, which helps to organize bookmarks and share them with others. His colleague has been using the application and has had some success organizing his bookmarks. He has convinced Daniel to install the application. Just after a tea break he is prepared to do this.

1. Daniel downloads the software and starts the installation process.

2. A setup program installs the application. Afterwards the extraction of knowledge from the local machine starts. The application accesses Daniels bookmark files and prompts him if he wants to share all of them. In combination the application displays a message regarding privacy issues describing how the sharing of bookmarks is done. Daniel is willing to share his bookmarks and communicates that to the system.

3. The application opens his bookmark files and tries to access all linked web pages. It classifies the links by the content of the retrieved web pages with the help of natural language processing and ontology learning.

4. A defined number (or percentage) of documents that could not be matched well by the application’s processing are presented to Daniel. He is now able to classify those documents manually. This step might also be omitted if Daniel does not wish to spend the time to organize the knowledge. One important part in this step is that the application provides a predefined topic ontology which had been created in the build step of the DILIGENT process. This helps users, who do not put much effort into organizing their resources, to retrieve resources easier than they used to do.

An unsolved problem is that new users will not know how the topic hierarchy is structured. Additionally empirical evaluation is needed to decide if the manual annotations of documents can be omitted or not, and how the threshold should be set that decides about which documents should be manually annotated.

An interval problem is that new users will not know how the topic hierarchy is structured. Additionally, empirical evaluation is needed to decide if the manual annotations of documents can be omitted or not, and how the threshold should be set that decides about which documents should be manually annotated.
6. Requirements Analysis: the BT Digital Library

5. Daniel can now choose if he wants to extend the topic hierarchy or if he wants to classify the bookmarks himself. Daniel chooses not to do so and leaves the classification totally up to the system. (For description of how manual classification could look like see “Add a Jotting” setting two and three.)

6. The application classifies the bookmarks according to the outputs of the natural language processing and ontology learning. Then the application finishes the installation process.

Additional remarks The network traffic for accessing websites can create a bottleneck and prevent Daniel from using his machine for a while. Furthermore, some web pages might not be available anymore. A default behavior for the application is not defined yet. Additionally it might make sense to introduce an advanced mode to allow more sophisticated users richer interaction with the application. Thresholds for the natural language processing and ontology learning could be changed by user in advanced mode. Additionally, more sophisticated possibilities to adapt the ontology may be needed.

6.6.2 Scenario “Advanced Adding a new Topic”

Intention of Scenario: This scenario describes a way for a user to add a new topic to the shared topic ontology. A user can create topics as well as a subTopic relation. In the scenarios “Add a Jotting” a user can only introduce a new topic, which is not connected in any way with other topics in the topic hierarchy. The here proposed functions give the user a more powerful tool for adapting the ontology to his own world view.

It is expected that this scenario will only apply to advanced users. Advanced users are represented by the persona Joe has been derived. Joe has technical experience, high IT skills and is a domain expert.

Expectations: Joe expects that he can modify the way documents are classified as well as the classification itself. Joe expects that he can extend the topic hierarchy as it is a vital feature for him as a researcher who is dealing with emerging technologies.

Setting: Joe has found an interesting web page which is related to the research project he is currently working on.

1. Joe is currently viewing this page. In his web browser Joe clicks the “add Jotting”-button.

2. The system processes the web page and suggests topics for Joe to apply.

3. Joe is not happy with those topics and tells the system he wants to change them.

4. The application enables Joe to browse the topic ontology to find an appropriate topic. Joe starts browsing the topic ontology, but he is not able to find an appropriate topic for the web page.
5. Joe decides that he wants to extend the topic ontology to suit his own needs. He chooses the direct super topic in the ontology and informs the application he wants to extend the topic ontology at this point.

6. The application prompts Joe to give a name for the new topic and if he wishes to give a short description of the topic. The application will now ask Joe why he introduced the topic. He is given a number of prototype reasons, but he can also choose not to give a reason as well as a different free text reason. Joe chooses one of the prototype reasons.

7. The application now prompts Joe if it should add the new Jotting to the given class.

8. Joe confirms this and the Jotting is classified against the new topic.

Additional Remarks: If Joe wants to create a completely new root element, he would not communicate to the application where he wants to extend the topic ontology. This is similar to setting 3 of the scenario “Add a Jotting”. In the setting described above it is not possible to create a new topic in between two topics, which are connected via the subTopic relation, nor is it possible to assign a different supertopic to a topic, thus reassigning parts of the topic tree to a different topic.

6.6.3 Scenario “Reorganizing the Knowledge Base”

Intention of Scenario: In this scenario the user can rearrange the topic hierarchy developed so far. The user might find it difficult to find a topic he is looking for. This can be due to the fact that he has created a topic which is too broad and he decides to split up a topic into several subtopics.

Expectations: Joe might experience some difficulties using the topic hierarchy because some concepts might not fit web pages very well, or a topic evolves and will need to be subdivided into several topics. Joe expects that this might be a rather laborious task. He might make up his mind as to what kind of concepts to add, before he starts.

Setting: Joe has been searching through his Jottings and found that he has problems finding Jottings that belong to a certain topic because he has a lot of documents that belong to one topic. Joe decides he wants to refine the topic ontology in order to retrieve Jottings faster.

1. Joe decides that he wants to reassign several old Jottings to a newly created topic.

2. Joe browses through the topic hierarchy to find the correct supertopic where he wants to extend the ontology and tells the application he wants to add a new subtopic here.

3. The application prompts Joe to give a name for the new topic and, if willing to provide a short description for the topic. The application will now ask Joe why he introduced the topic. He is given a number of prototype reasons. He can also choose not to give a reason or a free text reason.
4. The application prompts Joe to browse to the Jottings that should be removed from a certain topic to be added to the new topic. Joe starts to browse through the topic ontology and selects the appropriate Jottings and adds them to the new topic. The application shows Joe the other topics that a certain document belongs to, so that he can make a choice by being aware of the other topics.

5. Joe browse through the ontology for a last time to double check his decisions and finds that he is satisfied with his changes.

Additional remarks This scenario does not show every aspect of interaction necessary for extending the ontology. For an interaction designer extending the application it should raise awareness that it might be important for users to reorganize their knowledge base. This happens when either they have to change the topic ontology themselves or a new topic ontology is being deployed in the local update step of the DILIGENT process. In those cases users will need to adapt their organization of knowledge to the new ontology.

6.7 Conclusions and Summary

For the design of the prototype for searching and browsing the Digital Library and using Jottings only the first two sets of scenarios “search & browse” and “Jottings” will be used. The ontology extension scenarios used for the design prototype are covered by the scenarios on Jottings. The scenarios on “Advanced Ontology Adaptation” are included as a possible starting point for developing more sophisticated ontology editing and knowledge sharing functions.
Chapter 7

Prototype Design

In this chapter the final results of the prototype design are presented. As described in section 5.3, the prototype was designed in three steps. Firstly deriving the interaction framework in a paper & pencil prototype, then designing the prototype, which was afterwards evaluated. Finally the prototype was revised with the results from the evaluation. The revised prototype is presented here.

Due to time constrains the design could not be finalized, though all important principles of the interaction are shown, thus making a major contribution to the design of the application. For finalizing the design additional scenarios, so-called key path variant scenarios, like the design of a configuration page would be necessary. As a last step the prototype should be tested again with a small sample of the user group, thus ensuring that the redesign works for users.

In the following the prototype for both use cases “Search & Browse” and “Jottings” are described. Then a rough data model is described, which was developed during a brainstorming session with two other members of the SEKT project. How this data model evolves is described in a scenarios, which lets a user interact with the application. Finally in the last section open issues yet to be solved are described. Some of the issues are rather technical (and therefore out of scope), but nevertheless need further attention.

7.1 Description of the Prototype

“Search & Browse”

The description of the prototype on the use case “search & browse” is organized into the parts: general description, expected benefits, detailed description, technical issues, and potential problems. In the detailed description of the screens only issues are covered, that are particularly interesting on the single screen. Therefore the structure of the description differs in between screens.

7.1.1 General Description

The use case “search & browse” is intended to describe, how a user accesses resources in an ontology-based Digital Library. For an initial ontology the thesauri of the commercial ABI / Inform and the Inspec databases are used. These contain the

\[ \text{see 6.1} \]
two relations subTopic, and relatedTo. The relation subTopic is used to describe the fact that a topic can be subdivided into several other categories (or topics). The subTopic relation organizes topics in a hierarchy. The relation relatedTo is somewhat weaker. It organizes topics by creating a link between two topics to indicate a certain similarity. The challenge for the application is to reveal as much of the ontology as necessary, but to hide any unimportant part of it, thus reducing the complexity for the user.

The application uses the ontology for results ranking and disambiguating search results by query enrichment. The use of an ontology enables a user to find additional resources, which are semantically near to his search topic. A typical interaction with the system is comprised of the following steps. The user enters a search term, which will return a result set based on a simple keyword search. If the results are too general for the user, he is able to filter the results by choosing a topic from a list of topics, which have been used to annotate resources within this initial result set. This topic will then be added to the query with the Boolean operation ‘and’ (query enrichment). In the query interface this function can be reached if a user clicks on the link labeled “add”. If the user wishes he can also start to browse the ontology by clicking on one of the topics displayed on the results page. This opens up a box which displays subtopics and topics related to the topic a user has clicked on. If the user clicks on one of the topics displayed in the box a new query is started returning all resources annotated with the topic selected. This allows a user to navigate through the topic ontology, thus trying to find resources which are annotated with topics semantically near to his initial query.

The application also provides the user with a history of his searches, thus making it possible for a user to trace back his search path. As a foundation to understand user needs, the following three context scenarios were used: (i) “finding a specific resource”, (ii) “finding an unspecific resource on a specific topic”, and (iii) “finding an unspecific resource on an unspecific topic”.

7.1.2 Expected Benefits

By using topics instead of keywords for querying and results ranking, a user should find documents, which are more relevant for him. The initial keyword search allows hiding the very abstract top-level topics from the user and avoids listing a large number of topics of the topic ontology as entry points. With the user interface as it is right now the user is enabled to enrich his queries more easily. The application basically makes suggestions how a query could be enriched. In the approach at hand the user is able to choose which topics should be actually used for enriching the query, rather than the application making the choice.

7.1.3 Detailed Description of Prototype

This section visits the main functionalities designed for the use case “Search & Browse”. The screenshots shown are organized in a scenario exploiting all functionalities available.

The BT Digital Library Start Screen (figure 7.1)

Particularly interesting in this screen is the search functionality, which allows an initial key word search in the resources of the BT Digital Library, the users Jottings,
7. Prototype Design

and in other persons Jottings. In addition, an intranet search, a web search and a news search could be integrated using external partners. This would make the BT Digital Library the single point of entry for information searches and knowledge acquisition.

![Figure 7.1: The BT Digital Library start screen](image)

**Description of content area** In the main content area of the screen several personalized functions can be seen. “My Jottings” lets a user explore the Jottings he made with the Jottings functionality.

“My history” is the personal search history of a user. The three different symbols shown, identify the three different items that can occur in the history, namely “T” for a query using a (shared or personal) topic from the topic ontology, “D” for a document that has been viewed in the search process, and “S” for a query using key words. The “S” is used for indicating a search, because only a key word search will be received as a search by the user, retrieving topics or documents is a way of navigation from the user perspective. Behind the symbols the terms used for the query are listed, the for documents the publication date and the title are shown. The history function can be reached from every screen.

“My Journals” is a functionality that is already provided by the BT Digital Library as of today as well as “My Spaces”. “My Journals” refers to a section where a user can read journals he has subscribed to. The functionality of “My Spaces” will be enhanced by the use of the topic ontology. In “My Spaces” a user can subscribe to so-called information spaces. Information spaces collect resources from the BT Digital Library regarding a particular domain of interest. An information space is
7. Prototype Design

described using topics from the topic ontology. The exact design of this functionality is not clear at the time of writing this work.

In “Hot Topics” a user can see which queries are popular in the BT Digital Library. This function might allow to spot trends and understand which topics are important right now. To make this functionality more useful it might make sense to compare the most queried terms of the BT Digital Library with the personal profile of a user, only showing the topics which are of relevance for this user.

With the functionality “Other Jottings” a user explores the Jottings other library users have made. This is the knowledge sharing view on the Jottings, which allows users to see the Jottings other users have created. The number of people who have created a Jotting might indicate the relevance of Jottings. The main functionality is to see recently created Jottings and most popular URLs. The Jottings of other users are mainly available using the search functionality. In principal this functionality follows a similar design as used for del.icio.us.

Additional remarks For using all the above described functionalities the user needs to sign on except for the search. The privacy of a user regarding the Jottings is ensured by allowing anonymous user names.

Key Word Search Results (figure 7.2)

On the screen the result set extracted by a key word search using the two key words “bank” and “automating” is shown.

Left navigation bar On the left hand side of the screen the topics, which have been used to annotate the documents retrieved by the key word search are shown. The ranking of these topics is described in section 7.1.1. The topics allow the user to start browsing the ontology without having to deal with topics that are too general to understand. This approach follows the same “low threshold, high ceiling” approach as employed by QuizRDF [30].

The word “add” behind each topic allows the user to enrich his original query using the key words of his original search and the topic displayed in front of the word “add”. The behavior of the application if a user clicks on a topic is shown on the next screen in figure 7.4.

Behind the topics and the word “add” a number in brackets is shown, indicating how many resources are retrieved if a query is triggered. This allows a user to judge in advance what happens, if he clicks on one of the links.

The number of topics retrieved with the result set can be very large if some very broad terms are used in the query. Therefore only the most relevant topics are shown and more topics can be displayed by clicking on the link “More...”.

Description of content area At the top of the page the original search term is displayed in a text field, which can be used by the user to modify his search. By selecting the correct radio option this search box can be used for searching within the results set. Below the search box the number of retrieved resources is shown. By

\[2\] It needs be evaluated if the functionality should be triggered by a click or a mouse over interaction.
clicking on one the links the user can choose how many search records are shown per page. The links labeled “back” and “forward” allow navigating through the result set. All the functionalities on top of the page are also shown at the bottom of the page allowing a user to navigate more easily. At the bottom of the page the number of pages the result set contains is visible. The number of the actual result set page is shown also (e.g. page 2 out of 12). This is very similar to the design of the search Google engine.

Below the navigation functionality the actual result set is shown. In front of each record can appear two symbols depending, if the resource is available in form of a PDF or a text file. The title of each record is displayed as a link, which leads to a more detailed description of the resource as shown in figure 7.7. The author of a resource is also linked allowing directly a search for other works by the same author. In brackets follows the publication date of the resource and its data source. This should help a user to understand what kind of resource he has to expect. Below parts of the abstract are shown highlighting the original search terms.

**Right side bar** On the left hand side the advanced features can be opened up with a click on “advanced features” (see 7.3). Below the history functionality is displayed, which has been described above.

**Advanced Filtering Options of Searches (figure 7.3)**

On the screen the same situation is shown as on the screen before in figure 7.2, except for the additionally displayed advanced features in the right side bar.
7. Prototype Design

Right side bar  The advanced features allow the user to search for a publication being published in a certain time range. The years for restricting the search will need to be adapted by the beginning of a new year. The second functionality allows users to search for different publication types. If other resources with different publication types are introduced these need to be displayed, too (e.g. web pages and Jottings).

Figure 7.3: Advanced filtering options for searches

Browsing a Topic (figure 7.4)

The screen shows what happens if a user clicks on the link “bank automation”.

Left navigation bar: Ontology navigation box  If a user clicks on a topic a box opens up, which displays topics that are related to “bank automation”, thus letting the user navigate through the ontology.

Two types of relations between topics are shown to the user. The topics titled as “broader topics” and the topics titled as “narrower topics” are connected with the initial topic via the subTopic relation. The other relation shown is available through the topics titled as “related topics”, which are linked with the initial topic by the relatedTopic relation. Again are the numbers of results displayed (in brackets), which would be retrieved if the user clicks on the topic.

Important to notice is the description. It allows the user to unambiguously interpret what is meant by the topic he has chosen. This description will be generated automatically using natural language processing, though a manual correction might
be needed if the outcome has not the appropriate quality. The layout of this “ontology navigation box” with the broader topic at the top, the narrower topics at the bottom and the related topics on the same height as the initial topic should be preserved, as it indicates via the layout where the user navigates within the ontology.

Search Results for a Topic (figure 7.5 and 7.6)

If a user clicks on a topic, a query retrieves all documents annotated with this topic. This is exemplified in the figure 7.5 by the topic “bank automation”.

Principally the layout of the screen does not change except for the the topic description, which is now displayed. With the link at the end of the topic description the user can open up the ontology navigation box as shown in figure 7.6. The layout of the box stays very much the same mainly to ensure the consistency of the layout.

Data Record for a Resource (figure 7.7 and 7.8)

After a click on one of the titles in the result set the user sees all the annotations available for a resource.
7. Prototype Design

Figure 7.5: Search results for the topic “bank automation”

Figure 7.6: Navigating the topic ontology from the topic “bank automation”
7. Prototype Design

Figure 7.7: Results display for one document or more general resource

Left navigation bar  This navigation bar remains the same, when a user opens the data recorded for a resource, thus ensuring consistency.

Description of the content area  The title is shown, as well as the author. With a click on the author a new query is triggered querying for other works of the same author. Below the media types symbols are displayed indicating availability of the resource in that medium. Theses symbols and the media type name are linked leading to the actual resource. Then publication data is listed such as the journal and publication date as well as the source of the record. This is followed by the topics, which are used to annotate the document. If available also the free text terms an indexer\(^3\) used are displayed, which is not the case here. Finally the abstract of the resource is shown.

Ontology navigation box (figure 7.8)  With a click on a topic the user can open the ontology navigation box as shown for the example “bank automation”.

\(^3\)The records of the databases ABI / Inform and Inspec have a field, which allows the use of free text keywords for a more flexible indication of the content of a resource.
7. Prototype Design

Figure 7.8: Navigating the topic ontology from the topic “bank automation”

7.1.4 Technical Issues

Ranking of topics and results

The ranking of resources as search results and the ranking of the display of topics is not clearly defined yet. An approach based a relevance ranking and on personalization to bias the order of results displayed might be a good solution. The personalization could be achieved by creating a personal profile consisting of topics a user is interested in. This personal set of topics could then be compared with the topics within the query result. A resource would then be ranked higher if it appears in the user’s personal set of relevant topics. If this is not sufficient because only very few topics are within a user’s personal set of relevant topics, the semantic distance between topics in the query result set and topics in the user’s set of relevant topics could be measured. The shorter the distance the more relevant a topic would be.

Search engine syntax

For the interaction design it is also interesting to roughly describe a syntax that can be used to create advanced queries. This is interesting for users who are represented by the persona Joe. Generally the syntax allows to query against a certain part of the ontology. For some concepts a shortcut is introduced. Principally the query done with the construct `<qname:relationName>value`

---

4Semantic distance here means how many steps it would take to go from one topic to another, weighted by the different kind of relations used.

5It might turn out that the usage of the signs “<” and “>” creates technical problems and might need change.
thus allowing more intuitive names. These are:

- **author** for querying the author of a resource.
- **before** querying for a resource that was created before the specified year (e.g. `<before>1996`).
- **after** querying for a resource that was created after a certain year (e.g. `<after>1996`)
- **topic** for querying a topic
- **title** for querying a title
- **pubtype** for querying a certain publication type

To complete the syntax some additional definitions have to be made. A white space between two words is interpreted as an Boolean ‘and’, as can be seen in the search engine Google. Quotation marks such as ‘‘’’ are used to connect two terms as a phrase.

**Required technologies**

Several technologies required for this use case. Ontology Alignment is needed to integrate the two different thesauri of the ABI / Inform and Inspec databases. User profiling is needed for the ranking of topics and maybe the result sets of the keyword search. Natural language processing is employed for creating descriptions for the topics from the ABI / Inform and Inspec databases. The shared ontology derived from the thesauri could be refined using ontology learning.

**7.1.5 Potential Problems**

Two issues have been raised so far, which might be of a problem using the proposed user interaction design. The computational complexity for extracting topics from a result set of a key word search is rather high. This due to the current database structure. Although problem is not solved yet, there should be a way to circumvent it.

Another problem was raised during the evaluation. Users pointed out that the interface is rather complex and involves a lot of reading. It might turn out that the design proposed here is too complex for a user to grasp. The user might be overwhelmed by the choices he has. This problem might be solved by making a new visual design, thus leaving the interaction design as it is, but providing a clearer view of the application.
7.2 Description of the Prototype “Jotting”

In this section the design of the prototype for the use case Jottings is described. At first a general description of the functionality is given, then expected benefits are laid out, a detailed description of the prototype is given, required technologies are briefly visited, and finally some potential problems are described. As in section 7.1 in the detailed description of the screen only issues are covered, which are particularly interesting on the single screen. Therefore the structure might differ in between screens.

7.2.1 General Description

The use case “Jottings” describes a knowledge sharing application with functionality for organizing URLs in a new way, comparable to “bookmarks” or “favorites” in browsers, but more advanced. The advancement is done by providing a semi-automatic system to assign meaningful “topics” to URL’s, thereby improving the ability of a user to find a certain URL again. Additionally to a URL and topics, such a Jotting can have a description as well as a title.

A main concept of this application is to share Jottings with other users. The sharing of URLs can be done either actively by a user creating a Jotting and notifying another user, in an automatic way letting the application find other persons, who have an interest in the URL, or by allowing users to search through all the Jottings other users have made. Therefore the application takes a pull as well as push approach for the distribution of knowledge. Important for the here described design is that a user creating a Jotting has a self-interest to do so because he wants to find a web page again. The sharing is a side-effect for him and does not require any further altruistic action because he creates the Jotting anyhow.

Another important idea is the tight integration of the application into normal work tasks. The only user interface a normal user will see is a tool bar in his web browser, which allows him to create a new Jotting, edit or delete an existing Jotting, and navigate to other web pages that belong to the same topics as assigned to the last recalled Jotting. Additionally a user can search the digital library, his own Jottings, and the Jottings of others. The search for Jottings of others is integrated into a normal search within the BT Digital Library as described in the use case “search & browse”.

The described prototype is an example of how a software application, which supports the DILIGENT process can look like. The part of the process, which is considered here is the second step, called “local adaption”. This step allows a user to extend a shared ontology with his own, private concepts, which are in this case topics. In the Jotting tool a user can only introduce new topics consisting of a name and a description. He cannot introduce new relations between topics or reorganize the topic ontology. Though the functionality is not described in a scenario, a user is also able to edit and delete a topic he has created himself. The access to this functionality is realized using the topics in the browser toolbar.

A typical interaction with the application would be a user finding a web page he wants to recall at a later date. Therefore he decides to create a new Jotting for this page. In a pop-up window the user is able to edit the title and the URL of the web page. Additionally the user can enter a short description for the Jotting. As there are several thousand topics, the application uses natural language processing to select the topics that are most relevant. Some of these suggested topics are
7. Prototype Design

Figure 7.9: Topics of different users, which have Jottings pointing at the same URLs are candidates for a new shared topic.

proposed by the application to assign to the Jotting, some topics, which are less relevant, are just displayed, enabling the user to assign also one of these topics to the Jotting (this is done by two thresholds: the first indicating a selection and the second indicating to display a topic). All other topics are hidden from the user, but if a user wants to assign a topic not displayed he is able to browse the ontology. In the case a user wants to create a new topic, he has to search for this topic first to ensure there are no topics with exactly the same meaning. To give users a better understanding of topics, every topic has a short description, thus allowing to unambiguously understand what a topic comprises. If a user creates a new topic, he can choose to enter a description for the new topic.

Important to recognize here is that the topics a user creates are a potential input for the next steps of the DILIGENT process (analysis). The newly created topics indicate that some topic might be missing in the topic ontology and the DILIGENT board in the analysis step might consider to introduce a topic into the shared topic ontology. Especially if several Jottings of different users pointing to the same URLs use newly created topics, it is a strong suggestion to introduce a new topic. This new shared topic would have a similar meaning to the topics created by users (see figure 7.9).

To understand the needs and goals of users the following context scenarios have been used: (i) “Add a Jotting”, which explains how a user can create a new Jotting, (ii) “Use a Jotting”, which describes how a user retrieves a Jotting to navigate to a
7. Prototype Design

Jotting’s URL. For managing the Jottings the scenarios (iii) “Edit a Jotting” and (iv) “Delete a Jotting” are described. All scenarios are based on the persona Daniel.

7.2.2 Expected Benefits

The application should enable a user to organize links in a more efficient way. By making sure that the sharing is not an altruistic act the sharing should be more efficient. Users should want to use the tool because they see an instant benefit for themselves.

Additionally the functionality gives an example how the local adoption step in the in the DILIGENT process could be represented in a software application. It shows an unobtrusive way how an ontology can be extend by users for their own private world view. In the third and fourth step of the DILIGENT process (analysis and revision) the board will then review the private changes made in the ontology, trying to detect similarities between the users’ topics. These similarities are indicators for new topics that might be introduced to the shared ontology to better fit the needs of users.

These new topics could be used for trend spotting. For BT it is important to follow new trends in business and technological developments. New topics in the topic hierarchy might be a way to show if there is a new “hot” topic emerging. This new topic could then be presented to the users of the digital library whose personal profile contains other topics related to the new topic, which would indicate their interests.

7.2.3 Detailed Description of Prototype

This section visits the main functionalities designed for the use case “Jottings”. The screenshots shown are organized in a scenario exploiting all functionalities available.

Web browser with Toolbar for Jottings (figure 7.10)

In the figure a web browser is shown with an installed toolbar for creating Jottings. This toolbar allows a user to create Jottings with ease, not having to call a certain webpage or open up an other application. The use of a toolbar in a webbrowser allows a very tight integration in a users work tasks. The toolbar enables a user to create, edit and delete Jottings.

Description of Jotting toolbar  On the screen seen in figure 7.10 the button for creating a Jotting is active because the web page the user is viewing is not a recognized Jotting. If the URL of the current web page is recognized as a Jotting the user is able to either delete or edit this Jotting. The interface for the edit process is the same compared to the interface creating a new Jotting except for the fact that it is filled with the data from the already created Jotting.

If a URL is recognized as a Jotting the topics belonging to that Jotting are displayed in the toolbar, allowing the user to see these topics and navigate to other Jottings. This is achieved by opening up a menu and displaying the titles of other Jottings of this particular user. In this menu is also a functionality, which allows the user to edit or delete a topic he has created himself. The topics are displayed on the right to the create button.
Further right is a text box, which allows a user to search the BT Digital Library, his own jottings (“My Jottings”), and Jottings of other users (“Other Jottings”).

The functionality of the toolbar should be explored further as it could also allow navigation through the topic ontology by providing similar functionality as the above described ontology navigation box.

Creating a Jotting (figure 7.11 and 7.12)

Figure 7.11 shows the situation if a user has clicked on “Create” to create a new Jotting for that web page. A new window opens up with an DHTML front-end.

Basic Jotting informations (located at the top) In the window a text field is filled with the URL of the current web page, the text field below is filled with the title of the web page. The text field labeled “description” is empty, providing the possibility for a user to add his own description. The Jotting can be kept private if the check mark is removed by the users. The default is that the Jotting is visible for others.

Topic selection (located in the middle) An instruction for novice users is displayed explaining how to select topics. Below the application allows to assign topics to the Jotting. The application provides a sophisticated functionality for
users to select topics for the new Jotting. Natural language processing is used to create two types of suggestions for topics to assign. One set of topics is suggested as topics to be assigned to the Jotting. A second set of topics is displayed because it might also be relevant, but the outcome of the natural language processing was not distinct enough to propose those topics to be assigned. Topics that are computed to be likely a correct topic for the Jotting are shown as “selected topics” (also selected with a tick). Topics being not as relevant are displayed in the category “Other relevant topics”. The user can then decide, which topics should be really assigned to the Jotting by setting or removing the check mark in front of the topics.

The idea of using natural language processing should help the user to easily create a Jotting and assign topics from a topic ontology consisting of several thousand topics. A proof how well this idea will work in the work environment has yet to be made. It might very well be that the idea needs to be modified substantially. In addition to the topics filtered by natural language processing, topics that have been recently assigned to a Jotting are displayed.

If the user cannot find the right topic within the topics of the application he can search for this topic by using the search functionality at the bottom of the window. If a user wishes to create a new topic, he will always be forced to search for this topic first because it should be avoided that two topics with the same or similar meaning exist. In figure 7.13 the situation is shown when a user has searched for a topic. The application displays a second text field with the name of the topic the user searched for and a button allowing to create the new topic. If a user enters a different term in the text field for creating new topics, the application would at first search for this topic and then display the topic the user had searched for in this
text field. Now the user would be able to create the new topic.

The user can also browse his own topics to find the topic he wants to assign to the new Jotting. This is done by the button “browse my topics”. The screen, which opens up then is shown in figure 7.15.

The “done” button creates the Jotting and the “close” button closes the window without creating a new Jotting.

![Figure 7.12: Mouse over effect for revealing topic description](image)

**Topic descriptions** To disambiguate and explain the meaning of topics the user can find the description of the topic by moving the mouse over the topic in question. The description will appear in a box as in figure 7.12. The topic descriptions should also foster a common understanding of topic’s meaning.

**Searching for a Topic (figure 7.13)**

The figure shows the screen after the user has searched for a topic. All screen elements remained in the same state except for the “suggested topics” and the “recently used topics”. These are replaced by the search results. As it is very unlikely for a user to exactly find the right topic, topics could be described using key word vectors, though also a more sophisticated solution might be possible.
7. Prototype Design

Browsing the Topic Ontology (figure 7.14)

Figure 7.14 illustrates what happens when a user clicks on one of the linked topics, thus browsing the topic ontology. The application presents topics which are connected with a relation to the initial topic, being a broader, narrower, or related topic. Also the topic description of the active topic is (partly) displayed. Only topics that have a relation with at least one other topic are displayed with a link. Topics that do not have navigational path ways (being a dead end) are not linked. This will mostly, but not exclusively, concern topics created by a users.

Browsing Topics Created by a User (figure 7.15)

Figure 7.15 shows the result of the interaction of clicking on the button “browse my topics”. The topics are listed alphabetically allowing navigation by clicking on the letter A through Z.

Entering a Topic Description (figure 7.16)

If a user wants to create a new topic and clicks on the button “create new topic”, the screen shown in figure 7.16 appears. This screen enables the user to write a description for the topic he has created. The description should help the user to remember the understanding the topic. The user can skip entering a description by just clicking the done button with out entering a description.
7. Prototype Design

Figure 7.14: Browsing the topic ontology

Figure 7.15: Browsing topics created by a user
7. Prototype Design

Successful Creation of a New Topic
After entering a description and clicking on the “done” button the user will find the new topic under “selected topics” on the left hand side of the screen as shown in figure 7.17. The screen state is going back to the state it had been before creating the new topic.

Sharing the New Jotting (figure 7.18)
After clicking the “done” button the user is directed to a screen communicating the successful creation of a new Jotting (see figure 7.18).

Functionality sending a Jotting In addition to the creation of this new Jotting, the user can also notify other people about the URL he has found useful. For this a text field for entering e-mail addresses is available and a text field with a generated description of the URL. This allows the user to edit the message being send. For helping the user to send the message correctly to other people, he can access his address book, thus allowing to select other peoples e-mail addresses by using their names. The application then fills in the text field automatically the e-mail addresses. This notification function is optional and can be skipped by pressing the “close” button.
7. Prototype Design

Receiving a notification about a Jotting  The user receiving a notification about a Jotting form another person is able to chose, how he wants to receive this notification. Available choices are via e-mail, as instant message, or as a notification when he logs on to the BT Digital Library. These choices should allow an unobtrusive way of notifying others.

Browser Toolbar When a Jotting Is Detected

Figure 7.19 shows the situation after a Jotting for the webpage has been detected. The buttons edit and delete are now activated and the topics in the toolbar have changed to the topics assigned to the detected Jotting.

Retrieving a Jotting (figure 7.20)

To use a Jotting a user has created sometime ago the user can open the web page as shown in 7.20 from the BT Digital Library start page.

Left navigation bar  The topics that were used for creating Jottings for this particular user are sorted alphabetically and can be recalled easily using the linked letters A through Z. Additionally a search functionality is provided, searching only in the Jottings of the particular user.
7. Prototype Design

Figure 7.18: Sharing the new Jotting

Figure 7.19: Browser toolbar when a Jotting is detected
Content area  In the middle of the screen recently created Jottings are displayed. A data set for a particular Jotting consists of the title, the date it was created, how many other people have created a Jotting with the same URL, what topics have been assigned, and the personal description of the user. The topics are linked and allow the user to view all Jottings he has created using this particular topic. If the application knows that a URL is no more available, it displays the text “link broken” together with the Jotting. The links edit and delete allow the user to manage the Jottings.

Right toolbar  The right hand side of the window shows topics that have been recently used for Jottings. A click on such a Jotting would retrieve all the Jottings which are assigned to the topic.

Known Problems
The interaction design has one known problem that could not be resolved due to time constraints. The user is presented with suggestions for topics as in figure 7.11. These are presented in two different ways. One is set of topics is suggested strongly and listed under “Suggested topics”. The other part is listed under “Other relevant topics”. Additionally “Recently used topics are displayed. At the moment the user has no chance to go back to these “Other relevant topics” and “Recently used topics”. A button or link should be added, which allows the user to go back to the screen shown in figure 7.11 from the screens in the figures 7.13, 7.14, and 7.15.
7. Prototype Design

7.2.4 Required Technologies

For the use case “Jottings” the following technologies will be used. First and foremost natural language processing is used to find the correct topics, which are to be assigned to the Jotting. User profiling allows the distribution of new Jottings without the help of a human by providing topics that indicate interest in a Jotting. Natural language processing is also needed if a user searches for a particular topic. It is very unlikely that a user will know the exact name of a topic, therefore it is necessary to detect the semantic nearness between the user’s search term and the topics which are retrieved.

7.2.5 Potential Problems

Several issues have been raised regarding the prototype, which have to be solved for a successful implementation of the use case “Jottings”. One main concern is the use of natural language processing to analyze a web page for creating a new Jotting. It has already been pointed out that the idea of making suggestion for topics with the help of natural language processing is not a proven concept. Additionally the calculation time for generating the suggestion for topics might be too high to actually use the proposed interaction design. It might turn out that the web pages need to be processed in a batch job. Later on a user can then review the suggestions made. The quality of the suggested topics also might not be good enough to lower the effort for a knowledge worker to a necessary level to make the application useful. These problems might interrupt a user’s normal flow of work, which should be definitely avoided.

It might turn out that it is rather long winded to fill out all the fields provided in the application and selecting the correct topics. This might prove to be a major drawback for the proposed design because the goal to integrate the application tightly in users’ work tasks was in this case not achieved.

Finally the sheer size of the topic ontology (several thousand topics) might inhibit users to actually use the application in a good sense. It might be worth to create different topic ontologies for different business units. These topic ontologies could be matched to the shared topic ontology of the BT Digital Library. The introduction of these reduced ontologies would allow users to understand the “whole” topic ontology for their area, not being disturbed by topics which are irrelevant for them. This has the disadvantage that organizational boarders between different business units are not crossed with the topic ontology. Different business units could develop a different sorts of conflicting topics (see section 3.3). Introducing another layer between the overall shared topic ontology and the personal ontology of a user might allow to remove the natural language processing for assigning topics proposed above. A user might be able to classify documents just by choosing topics form a business units topic ontology.
7.3 Rough Data Model

This section describes a rough data model for the use case Jotting. A brief comparison of the here proposed data model with the data model of Annotea (see section 3.6 and [41]) is given here. The data model was developed in a brainstorming session together with two SEKT project members. The topic of the brainstorming was to understand the implications of the context scenarios on the application.

In the following the design is of the data model is described with the help of a scenario. The personas Daniel and Joe use the application to create new Jottings. The scenario focuses on technical aspects rather than on the user experience. Figure A.1 below shows how the topic ontology is extended by the personas' actions. A description of the steps follows. (The steps described below can be found using the numbers in the figure.)

Daniel creates a Jotting with a URL and enters his description. None of the topics suggested by the application fit the Jotting, so he creates his own new topic “SEKT”. He does not choose to keep the Jotting private.

1. The application creates the Jotting (J1) with the URL and extracts the keywords (“XML”, “JAVA”, and “SUN”).

2. The application creates JottingPerspective1 (JP1), using the extracted keywords (“XML”, “SUN”, “JAVA”).
   (a) With these keywords the application maps JP1 to topics in the shared Digital Library topic ontology to the topic “XML”.

3. The application creates JottingPerspective2 (JP2) using the extracted keywords and the description Daniel entered.

4. The application creates a new topic UserTopic1 (UT1) with the extracted keywords of J1 using the name “SEKT” Daniel had entered. The keywords for this new topic are now “SEKT”, “XML”, “SUN”, and “JAVA”. The topic will be displayed with the name “SEKT”. If the user did not enter a topic description a summary of all documents belonging to the new topics is generated to make a description for the topic available.

5. The new topic “SEKT” is added to Daniel’s profile.

6. Daniel finds another web page he wants to remember, so he creates a Jotting for this web page, using the topic “SEKT”.
   (a) The application creates a new Jotting (J2), including the URL and the extracted keyword (“WEB”).
   (b) The application creates JottingPerspective4 with the owner being Daniel, using Daniel’s description, and setting its visibility to public.
   (c) The application aids Daniel to add JottingPerspective4 (JP4) to the UserTopic1 (UT1) based on similar keywords found in the web page and in the keyword bucket of UT1. The application adds the keyword “WEB” to UT1.
   (d) The application creates JottingPerspective3 (JP3), with the extracted keyword (“WEB”) and the library as owner.
7. Prototype Design

(e) The application maps JP3 to topics in the shared Digital Library topic ontology, using the topic “SUN”.

7. The persona Joe copies the Jotting J2 from user Daniel. The only thing he changes is the topics describing the Jotting for Joe. He uses his own topic “PROJECT” and the shared topic “SUN”.

(a) The application creates the new JottingPerspective5 (JP5). The only extracted keyword is “WEB”, and the names of the topics entered by Joe “PROJECT” and “SUN”.

(b) The application creates the new topic UserTopic2 (UT2). The topic will be displayed with the name “PROJECT” and has the extracted keyword “WEB”, and the topic name entered by Joe “PROJECT”. Additionally the topic “SUN” is assigned to the Jotting, as Joe has told the application to do so.

(c) The topic UT2 is added to Joe’s profile.

The indirection using a Jotting perspective is necessary to allow different views on the same Jotting. By allowing different views on the same Jotting, it is possible that different users can assign different topics to the same Jotting. Therefore different users can create differing world views using the same Jotting. The use of the DILIGENT process might eventually merge these differing world views into one, when a new updated shared ontology is distributed (fifth step of the DILIGENT process “local update”).

A keyword bucket for Jotting perspectives is not necessary. It might only be helpful for retrieving a Jotting, when a user searches for it. Usually this could also be done by using the keyword vectors of topics and Jottings.

The descriptions of the topics are an important help for a user navigating through the ontology and assigning correct topics to Jottings. This is feature should not be underestimated because it is the way a user can tell, which topic he should use, thus disambiguating topics. A topic’s name cannot be enough for the disambiguation, as it does not necessarily contain enough semantic information for a user to assign it in a correct way.

In the data model described above one relation between topics is not mentioned. The only two relations that are important from the perspective of the front-end are subTopic and relatedTopic. Their use has been described in section 7.1.1. A sophisticated data model for the BT Digital Library is being developed within the SEKT project. This data model takes into account the data model of all sources the BT Digital Library is using, such as the Abi/Inform and Inspec data bases and others. As of the knowledge of the author no specification document has been finalized yet.

The here proposed data model might turn out to support the automatic suggestion of topics better than the data model used in Annotea as it provides possibilities to store key word vectors. Also the idea of a JottingPerspective introduced in the here proposed data model cannot be found in the Annotea data model. Noteworthy is the ability of the Annotea data model to store data about metadata (Shortcuts).
7.4 Other design issues

Right now a configuration tool for the application has not been designed though it is necessary for both use cases. For “search & browse” the configuration would allow a user to adapt his personal profile, which consists of different topics and for the use case “Jotting” it is necessary because the user needs a place to configure how he wants to receive notifications about other peoples Jottings.

Interesting is the granularity of topics. The question must be raised how detailed the topic ontology should be. It might prove useful to refine the topic ontology provided by the Abi / Inform and Inspec thesauri. This should have the effect that a query for one topic would retrieve less documents, thus allowing a user to get an overview about a complete topic. The disadvantage with a more fine grained topic ontology is the maintenance cost. It might as well prove that the costs are prohibitively high constructing and maintaining a refined topic ontology.

Another problem that is not solved because it is less a problem of interaction design, is how to deal with a resource that is stored in more than one data sources. There needs to be a sophisticated solution to detect those duplicates to avoid displaying the same item more than once. The use of topics should help to achieve this task.

Not only duplicates are a problem but also the ranking of hits form different data sources during a search need to be included in one sophisticated ranking scheme. Again should the use of the topic ontology be very helpful because the semantic nearness of of topics can be used.
Chapter 8

Evaluation of Prototype

The proposed design for a prototype application has been evaluated twice. Following the development of the interaction framework an informal evaluation with potential users took place. The users were partly connected to the SEKT project, which could bias their view, and were also technical experts. For the evaluation a paper & pencil prototype was used.

The second evaluation was complete on a more formal basis. Interviews of about an hour length were conducted with users. During the interview the users were shown the prototype in form of a Microsoft PowerPoint presentation. While viewing the prototype the users were asked to “think out loud” to express what they see, expect, and what surprises them. At the end of the interview the users were asked to answer eleven questions (documented in the appendix under C.3). The prototype shown in section 7 has been revised since the user study. A third user study was not possible because the author did not have access to users at BT anymore.

Five users were recruited for the second evaluation. Two of the five users had never used the BT Digital Library before, two users had used the BT Digital Library once and one user was working in the team of the BT Digital Library and therefore used it often. Four of the participants have engineering-based jobs and one has a managerial job.

The results of the evaluation are listed below. Generally these results allow to resolve design issues, but cannot show if the application is really useful as a knowledge sharing application. This is partly due to the fact that it was difficult for users to imagine how the final application would really work. This understanding of how the application works was not fostered by the design of the scenarios the user were shown during the interview. The scenarios included a lot of functionality, rather than focusing on showing what a typical interaction with the system would be like. As of the tight deadlines it was not possible to design a more thought through interview, with clearly defined goals, providing guidelines for asking each user similar questions about the prototype design.

The listed action items are concluded from the user study of the PowerPoint prototype. The action items have been considered in revising the prototype shown in chapter 7. Additionally the description of the prototype has been influenced by these action items. The prototype used for the user evaluation can be found in the appendix (see sections C.1 and C.2).

The presentation of the results is divided into comments on the use case “search & browse”, comments on the use case “Jottings”, and the results of the question-
8. Evaluation of Prototype

The major findings are described, which have lead to the redesign of the prototype as it is shown in section 7. Finally an interpretation of the results is provided.

8.1 Results on the Use Case “Search & Browse”

The summarized user comments are shown in normal print. Additional comments of the author are printed in italic.

General comments:

• The description of a topic is important if a term is not known to a user.

• If a query for a single topic is submitted, a description of that topic in the result would be helpful.

• Giving the choice to display 100 results is good, because if the page loads fast, one can have a quicker scan.

• Users commented that the complexity of the screens is rather high because it is full with text and it requires effort to read the text.

• Users commented that the term “my spaces” is not very clear. The whole term “information space” might be better for orientation or something like “interest group”. This was mentioned more than once.

• The example “bank automation” used in the prototype is poor because a key word search compared to a topic search retrieves different results. The difference was hard to grasp for interviewees.

• An additional publication type could be white papers. This would be an extension of the Libraries resources.

Advanced features:

• “Published in” in the advanced features was unclear for a user. This might just be due to the laziness to read thoroughly.

• More information (affiliation, biography, etc.) on authors would be useful. By clicking on the authors name not only a query for all known publications could be triggered but also this additional information.

• The way how to search for physical books was unclear. The design envisions that books will be retrieved together with the other resources.

Navigation bar left on the hand side:

• The navigation is usually on the left side, the filtering for results is somewhere else. This inconsistency creates a problem for the understanding of what is happening.
8. Evaluation of Prototype

- A tree view navigation of the relevant topic ontology was suggested (comparable to a file system) by a user. This was mentioned more than one user. One user suggested that it might be helpful to switch between a hierarchical and alphabetical way of displaying relevant topics. Another user commented that the hierarchical way of displaying the ontology might be more complex, but also better. Introducing the tree navigation means that the complexity of the user interface increases and the user has to make more effort to explore the available options. An alternative might be to order the topics by relevance. This would reduce the overall complexity of the user interface and reduce the effort to find relevant topics for a user. The switching between hierarchical and relevance or alphabetical order also puts an additional option on the screen, which might be critical, as the user-interface is already quite complex.

- The link “add topic”, for adding a topic to a query, was understood by one user as adding a topic to the personal profile. Other users were also not sure what “add topic” means. Even though this functionality was unclear for user, it will continue to be in the application because it is thought to be a useful feature, although yet unknown to users. It can be easily understood by using it. Especially problematic for users was that the “add topic” would add a search term to the query connecting it with and Boolean ‘and’. The idea of using this should be evaluated further in a functional prototype.

- The term “add topic” does not make sense for broader or narrower topics. It just makes sense for related topics (make boolean intersection between two related topics). Important is that the annotation of documents with the topics might need further modification. A document might have been classified with a topic in the same subTopic tree, being a subtopic of another topic used as an annotation. This should not happen, as it is an inconsistency in the data structure. An example of this is if “Bank Automation” is a subtopic of “Automation” only one of the topics should be used as topic, but never both. The data set needs to be checked to comply to these rules.

- It might be hard to grasp for users that links in the ontology navigation box, which opens up by clicking on a topic, are not filtering the results, but switching the topic. For one user the ontology navigation box is not really necessary. Contrarily, another user commented that the navigation box is helpful. It is one of the main principals the interaction design follows. Users opinions show that no clear conclusion could be reached on wether the ontology navigation box is really helpful.

- A user commented that alternative topics are generally useful for browsing.

History:

- The history for more than one session might not make sense. An overall search history could be switched on or off in preferences. The feature might still be useful for other users. Further evaluation is needed to decide how this feature should be implemented.

Result sets for search and for one resource:
8. Evaluation of Prototype

- Topics should be visible on the initial results page (after the first query, not after selecting a single document). The abstract would not make sense to display on this results page as it is too long.

- On the data record screen the display of topics, which are only relevant to the document itself in the left-hand column, is inconsistent because it changes compared to the initial search screen. The topic navigation on the left-hand side should not change too often because it should follow the normal behavior of a navigation on web pages.

- It would help to show the page number of the result set page a user views at the moment.

Syntax for search engine:

- The syntax for searches should be close to Google. Quotation marks for creating a phrase like topic: “bank automation” should be used.

Other issues on “Search & Browse”:

- Filtering and moving around in the topic hierarchy is not clearly separated e.g. what happens when a user clicks on a topic? Is it a filter on the result set or does he start a new search? This needs to be made very clear.

- Orientation (where am I?) is still weak for users. History needs to be revised, and an orientation of which topic is “active” should be shown.

- Advanced features need a “search”-button.
- Advanced features might need to be renamed.

Item was discarded.

In the following action are listed, which are concluded from the users’ comments. The action items were the basis for the last revision of the prototype.

Action Items:

- Make the distinction clear between filtering search results and moving in the topic hierarchy.

- If the user queries for a single topic, the topic description needs to be displayed on the result set page.

- The page for displaying the data record for one resource has on the left hand side the same navigation view as the initial results set page for a query. The topics describing a resource will need to be placed differently.

- Define a little bit of syntax for the text field of the search engine. This has been done in chapter 7.1.1

- Display the number of the page in the result set (e.g. page 1 out of 10).

- Add search button for advanced features.
8. Results on the Use Case “Jottings”

The user comments are shown in normal print. Additional comments of the author are in printed in italic. Later on the action items concluded from the user comments are listed.

8.2.1 User comments

Navigational issues:

- A “back”-button is missing. This was mentioned twice.

- A history is missing in search for topics. The features might be useful, but it needs further evaluation to judge the real relevance. It might also be additional overhead.

- Link “browse my topics” in Jotting pop-up is inconsistently represented because it is similar to creating a new topic or searching for a topic. It should be a button and maybe have a short description of what will happen, if a user clicks on the button. Additionally, a user commented that the position is not thoughtfully chosen.

Wording: Generally the wording use in the design needs to be double checked by a native speaker.

- The naming for the functionality “Jotting” is very unclear especially for none-native speakers.

- The expression “Matching topic” is imprecise. This functionality needs to be restructured. The application should suggest a couple of topics.

- The word “share” for sending a message about a Jotting to another person is imprecise.

- Distinction between “all topics” and “my topics” is unclear. It needs to be clear to the user where other users’ Jottings are. This concerns the screen shown in C.21. The link needs to be removed because the Jottings of other users are available via the normal search interface.

- Replace done with save on the button. Done should be replaced as either “next” or “save” depending on the screen and if another interaction follows. This item was discarded.

- The difference in between “matching topic” and “active topic” is unclear. The issue was mentioned by more than one user. The design idea was removed from the application. A “matching topic” is a topic that exactly matches the users search and is therefore displayed as in figure 7.14. Otherwise a normal results list is displayed as in figure 7.13.

- “Recent Jottings” was understood as Jottings recently visited and not Jottings recently created. Basically a history was expected. [Item discarded.]

- Distinction for a user between “my topics” and “my Jottings” was unclear. [Item discarded]
8. Evaluation of Prototype

Layout inconsistencies:

- Location of selected topics is badly selected, this was mentioned twice.
- Search box for topics is not in a consistent place. This was mentioned more than once.
- Restructure page with narrower and broader topics. The dividers between narrower/broader/related topics are irritating.
- The text field for entering the description of topics needs a clear description of what is expected. Additionally the name of the new topic is not very prominent on the page. Assure the user knows entering a topic description is a normal part of the process creating a topic. The name of the new topic should be more prominent on the page.
- The “send” Button needs to be clearer in the context of the text field. The wording should be “send and save”. The button should additionally be nearer to the text field.
- The functionality “Search or create a new topic” is not placed very prominent. This might not need change because a user should not need the feature very often.

Display of created Jottings:

- The navigation to a topic in the display of Jottings is inconsistent with the design of the use case “search & browse”. “Search & browse” displays recent topics on the right-hand side. The topics for navigation are on the left-hand side.
- The header is missing on the web page for Jottings. This is also missing in other parts. The comment will be discarded because it does not show any importance for the interaction with the system.

Suggested additional features:

- A user missed the functionality to create relations between topics, delete or edit topics. The functionality was not missed by all users. It might also be a feature that will be rarely used by users. Additionally it would increase the complexity of the user interface. Nevertheless it might be a useful feature.
- When a user gets an e-mail with a Jotting of another user he might want to add a newly created topic to his topics. Maybe this topic could be created automatically. It would be interesting to watch the effects of that regarding the DILIGENT process. This could help to build similar topics. But in the envisioned way of how to represent Jottings right now, a new topic would be created as a personal topic of the user receiving the e-mail. Another issue is that the user might already have a similar topic, with a different name. The application would need to do a mapping between the new topic of the user sending the Jotting and the user receiving the Jotting.
- It might be interesting to cache the content a URL is pointing at for later retrieval. To mark interesting sections might be useful. A problem occurs with the copy-right. It might not be permitted to store the page.
8. Evaluation of Prototype

Feature, which could be removed:

- A user commented that it is not necessary to display already selected topics when searching for an additional topic. This needs further evaluation as the feature might still be useful for others.

- Recently used topics are not always necessary. This also needs further evaluation. It might still be a useful feature for other users.

Additional descriptions might be needed:

- The search in topics when creating a Jotting is unclear. In which set am I searching? What is being queried?

- MouseOver for ontology navigation box might need a description. Maybe this would not be the case if a user would have seen a real application compared to the prototype.

- It is unclear why certain topics have a tick set and some don’t. This needs further explanation, but could also be not an issue if a user actual works with the application.

- Visibility of Jottings for other users is unclear.

General issues:

- One user expected that topics are created globally. It should be made clear that topic are only “private”. The wording could be “private” or “your own topic”.

- The automatically generated messages for pushing a Jotting to others should be editable.

- In the browser toolbar for Jottings the buttons such as “create Jotting”, “delete”, and “edit” should always be visible, but marked as deactivated, if not available.

- MouseOver boxes with topic description shouldn’t be too big.

Other issues on “search & browse”:

- The orientation for a user, who creates a Jotting, is weak during the search for a topic.

- What happens if the topic is a dead end? Describe that there are no links if a topic is a dead end. This is especially important for “my topics”, which cannot have relations right now.
8.2.2 Action Items

In the following actions concluded from the user comments are presented. The items have been completed during the last revision of the prototype, if not indicated differently.

- Navigation is weak in Jottings:
  - Add a back button.
  - Where am I? It needs to be shown clearly if a user is creating a Jotting or a topic.
  - Change “Browse my topics” link to a button more clearly indicating the same functionality as search and create topics.

- Wording:
  - Replace “share” with “send to”.
  - Replace “done” with next or save depending on the screen. *Omitted in redesign.*
  - Make a clear and obvious distinction between active and matching topics. *Discarded because screen of matching topics was removed*
  - Make clear that new topics are private, but visible for the board. *Omitted in redesign.*

- General interaction:
  - Create screen for the case if there are some choices for a topic which might match a search for a topic.
  - Provide description of how Jottings are public or private. *This has been changed to providing a better label for the check box deciding if a Jotting is private.*
  - The active topic should have the description of that topic showing on screen.

- Layout:
  - Change position of selected topics.
  - Put text field for searching topics at a consistent place.
  - Restructure page with broader, narrower and related topics. Change especially the white separator bars.
  - Page for topic description needs to be more consistent with the other pages.
  - Change page of results display of created Jottings. Make it more consistent with the use case search & browse.
  - In the toolbar create, delete, and edit should always be visible.
8.3 Results of the Questionnaire

In this section the results of the questionnaire, which was used during the user interviews, are described. The first two questions regarding demographical data have already been described above (see 8).

The term Jotting was understood only by one user, who had used the implemented Jotting functionality of the BT Digital Library and was the only native speaker within the studied user group. To all other users the word “jotting” or “to jot” was unknown.

Users were generally careful on saying if the design for search & browse would be better or not. Generally they were positive, but unsure if the used design was an improvement.

The prototype used for the user evaluation showed two possibilities for ranking the topics in the initial results screen. One was using a (not thought through) clustering method and the other was just ordering the topics alphabetically (see figure 7.2) on the left hand side of the screen. The comments of users suggested that the “cluster” solution was liked more, although it was confusing. Only one user liked the alphabetical order better. Comments on how to improve the design have lead to the creation of a new solution, which ranks the topics by relevance to the query and the users personal profile (see section 7.1.3).

The usefulness of the design for the use case “Jottings” was received more diverse than for the use case “search & browse”. One user is already using the functionality implemented in the BT Digital Library. One user was skeptical if it would help him organize his bookmarks, two others thought it might help them. One user commented that he dislikes central solutions because they might close down and he would loose his data. This could be avoided by providing a back-up function, allowing the user to download all his Jottings.

Users thought they would be able to classify the bookmarks with topics. One user commented though that the mass of topics (several thousand) might prevent a classification.

The majority of users would like to create relations between topics (4 out of 5). They commented that it might not always be necessary and some users might not want to create relations. One user did not want to create relations, himself.

Generally users wanted control over the topics they had created, thus editing and deleting these topics.

One user commented that different browsers should be supported using the Jotting toolbar. Another user commented that the user interface was displaying a lot of text, which was overwhelming for the user. Having some place for copying and saving important parts together with the Jotting, was missing for yet another user. Also collecting news feeds would be seen as good service by this user.

8.4 Findings of the Evaluation

Although users commented cautiously positive about the features of the application no clear conclusions can be reached on whether the application would seriously improve organizing bookmarks as well as searching and browsing in a digital library. In order to reach a conclusion, a better prepared user study would be necessary. If this cannot be done soon within the SEKT project, it should be done at later date, when an implemented prototype exists. Positive is the comment of a user, who
uses the Jotting function of the BT Digital Library, indicating that the principal of Jottings is very useful.

Generally the user study was very helpful for resolving design issues because it provided the author with differing views on the application. Many design issues could be resolved in the revision of the prototype. The action items which were concluded from the user study have already been listed.

Two results of the questionnaire were especially interesting. Generally, users wish to have control over the usage of topics. They want to be able to edit and delete topics. In addition they want to be able create relations between different topics. This shows that, in a more sophisticated solution, these issues should be tackled.

The other interesting result of the questionnaire is that users felt definitely able to classify the webpages to topics. This might be due to the fact that users did not know that several thousand topics are available for the classification. The user from the BT Digital Library commented he would not be able to classify the results according to the ABI / Inform and Inspec thesauri. A further study on how well people are able to classify webpages according to (differing numbers of) topics would be interesting to conduct.
Part III

Conclusion
Chapter 9

Future Work and Conclusion

This section provides an outlook on possible future work and summarizes the results of the work at hand to draw a conclusion.

9.1 Future Work

In this section ideas for further work are described that was highlighted during the design process. The use cases “search & browse” and Jottings are viewed separately.

Search & browse  In the future the use case search & browse might be advanced using the paradigm of “zoomable interfaces” as introduced by [24]. This would allow users to organize the found resources for reading.

If more dimensions for searching are available the idea of Hearst, proposed in [37] using different kinds of topic hierarchies to search for the same resources might be worth further investigation. The data structure of the thesauri prevented a reasonable use.

In addition to this it should be thought of how (literature) references in the resources could be used to enhance the browsing experience. A possible scenario might be to display links to the resources cited and to the resource that cite another resource.

An approach, which has not been explored in the user interface proposed here is to offer users synonyms of their query for query enrichment. A key word search might be improved substantially if synonyms could be added to the search. The results ranking though would be more important in such a search because more (irrelevant) resources would be retrieved.

More sophisticated functionalities for managing and storing different searches may enhance the use case “search & browse”. It might be interesting for a user to store different searches he has done regarding a certain topic. This might also help to detect a search context more easily, thus helping to create more sophisticated focused crawling mechanisms.

For further development it would be interesting to merge search results from different information sources such as intranet, Internet, Digital Library and the knowledge on a local machine. This could be done using a peer-to-peer client.

Additionally the initial searches could improve the retrieval of more relevant results by trying to detect, if the retrieved topic is of interest for a user or if he
has to navigate the ontology to find the right topic. If a user has to navigate, the keyword used for the query could also be assigned to the topic the user navigated to, in case that happens to a number of users. This could make it possible to evolve the ontology (compare also [62]).

**Jottings** The use case Jottings could be extended especially with a more sophisticated way of editing the topic ontology. The user study shows that users want more control over topics, editing and deleting topics, as well as creating relations between topics. Important for this is that the interaction needs to be very easy to carry out and to understand the effects of changes. Also the interaction possibilities of DHTML might not be sufficient for more sophisticated ontology editing. It might prove useful to employ a web browser plug-in for the editing of ontology such as Java applets or Macromedia Flash.

The use case “Jottings” might hold some promises also sharing other resources than just URLs. Important for this though is the tight integration into user’s work tasks. If one is writing a document with a word processing application, the application might ask the user to publish the document when he wants to save the document. This might create a problem because drafts should not be published. Also problematic is extending the word processor to support this task. Implementing the extension might turn out to be a laborious task, which continuously needs to be ported to new versions of the word processor.

### 9.2 Conclusion

The prototype for “search & browse” continues to develop ideas from QuizRDF on how users can interact with a large ontology searching and browsing to find relevant resources. The “low threshold, high ceiling” approach QuizRDF uses has been developed to be more user-friendly providing better orientation and a more sophisticated way of browsing through the ontology. Some ideas on using personalization and the ontology for query enrichment have also been introduced. The ontology navigation box should help users to move through the ontology in a sophisticated way. Generally the use of well-defined topics allows a user to disambiguate his initial query. For the understanding of the topics the user can recall a brief description of the topic. Also the use of a controlled vocabulary should improve search results. Generally key word searches will continue to be an important tool for information retrieval because they allow users to search for words out of the original context. A search in metadata only provides access to the data from the perspective of the describing metadata. Therefore an approach as proposed in this work uses the best of both: simple key word searches as well as well-defined semantics for disambiguation and query refinement.

The prototype for the use case “Jottings” proposes a knowledge sharing tool, which is tightly integrated into users work tasks. An important factor is that the sharing is not an altruistic act for the knowledge “seller” (as described by Davenport and Prusak in [28]). This knowledge sharing application makes sophisticated use of topics as a means to assign well-defined meaning using an ontology. New topics can be introduced by a user, thus showing how an application can support the local adaption step of the DILIGENT process in a simple way.

The approach of using Goal-directed Design for designing a knowledge management system seems to fulfill the goal of a tight integration into the work task of users
well. The literature on knowledge management especially mentions that only a very small amount of knowledge management is actually manageable with IT-systems, saying that knowledge management must focus on human issues. Therefore the author is convinced that the use of a human-centered design methodology is essential for designing knowledge management systems. This approach has been taken in this thesis, in contrast to a design methodology that is only focused on the underlying technology. The usage of Goal-directed Design, especially in the design of the prototype, allowed to focus on the goals of the future users, rather than on using as much technology provided by the SEKT project. Goal-directed Design made it possible to design an application, which makes appropriate use of the technology developed by the SEKT project.
Part IV

Appendix
Bibliography

   . Amaya - W3C's experimental Editor/Browser.


[5] Namespaces in XML.

[6] DAML+OIL.
   2005.


[8] Defining N-ary Relations on the Semantic Web: Use With Individuals. Internet,
   July 2004.
   http://www.w3.org/TR/2004/WD-swbp-n-aryRelations-20040721/.


    028-istifp6-november04.pdf.


[12] Andreas Abecker and Ludger van Elst. Handbook on Ontologies, chapter On-

[13] David Alsmeyer, Allyson Cheung, Michael Engler, Nick Kings, Ian Thurlow,

[14] David Alsmeyer, Allyson Cheung, Nick Kings, Faye Skinner, Ian Thurlow,
BIBLIOGRAPHY


# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Posting a URL using del.icio.us.</td>
<td>19</td>
</tr>
<tr>
<td>2.2</td>
<td>The Jotting functionality of BT Digital Library</td>
<td>19</td>
</tr>
<tr>
<td>3.1</td>
<td>The figure shows the architecture layers of the semantic web. [40]</td>
<td>23</td>
</tr>
<tr>
<td>3.2</td>
<td>An example for a simple topic ontology.</td>
<td>26</td>
</tr>
<tr>
<td>3.3</td>
<td>The DILIGENT process</td>
<td>28</td>
</tr>
<tr>
<td>7.1</td>
<td>The BT Digital Library start screen</td>
<td>71</td>
</tr>
<tr>
<td>7.2</td>
<td>Search results for the the key words “bank” and “automating”</td>
<td>73</td>
</tr>
<tr>
<td>7.3</td>
<td>Advanced filtering options for searches</td>
<td>74</td>
</tr>
<tr>
<td>7.4</td>
<td>Click on link for the topic “bank automation”</td>
<td>75</td>
</tr>
<tr>
<td>7.5</td>
<td>Search results for the topic “bank automation”</td>
<td>76</td>
</tr>
<tr>
<td>7.6</td>
<td>Navigating the topic ontology from the topic “bank automation”</td>
<td>76</td>
</tr>
<tr>
<td>7.7</td>
<td>Results display for one document or more general resource</td>
<td>77</td>
</tr>
<tr>
<td>7.8</td>
<td>Navigating the topic ontology from the topic “bank automation”</td>
<td>78</td>
</tr>
<tr>
<td>7.9</td>
<td>Topics of different users, which have Jottings pointing at the same URLs are candidates for a new shared topic.</td>
<td>81</td>
</tr>
<tr>
<td>7.10</td>
<td>Web browser with toolbar for Jottings</td>
<td>83</td>
</tr>
<tr>
<td>7.11</td>
<td>Creating a Jotting</td>
<td>84</td>
</tr>
<tr>
<td>7.12</td>
<td>Mouse over effect for revealing topic description</td>
<td>85</td>
</tr>
<tr>
<td>7.13</td>
<td>Searching for a topic</td>
<td>86</td>
</tr>
<tr>
<td>7.14</td>
<td>Browsing the topic ontology</td>
<td>87</td>
</tr>
<tr>
<td>7.15</td>
<td>Browsing topics created by a user</td>
<td>87</td>
</tr>
<tr>
<td>7.16</td>
<td>Entering a topic description</td>
<td>88</td>
</tr>
<tr>
<td>7.17</td>
<td>Successful creation of a new topic</td>
<td>89</td>
</tr>
<tr>
<td>7.18</td>
<td>Sharing the new Jotting</td>
<td>90</td>
</tr>
<tr>
<td>7.19</td>
<td>Browser toolbar when a Jotting is detected</td>
<td>90</td>
</tr>
<tr>
<td>7.20</td>
<td>Retrieving a Jotting</td>
<td>91</td>
</tr>
<tr>
<td>A.1</td>
<td>A rough data model for the use case Jottings</td>
<td>128</td>
</tr>
<tr>
<td>B.1</td>
<td>Paper &amp; pencil prototype: Start screen of the BT Digital Library</td>
<td>130</td>
</tr>
<tr>
<td>B.2</td>
<td>Paper &amp; pencil prototype: Result set of a search</td>
<td>131</td>
</tr>
<tr>
<td>B.3</td>
<td>Paper &amp; pencil prototype: Result record for one resource</td>
<td>132</td>
</tr>
<tr>
<td>B.4</td>
<td>Paper &amp; pencil prototype: Advanced search features</td>
<td>133</td>
</tr>
<tr>
<td>B.5</td>
<td>Paper &amp; pencil prototype: Creating a new Jotting</td>
<td>134</td>
</tr>
<tr>
<td>B.6</td>
<td>Paper &amp; pencil prototype: entering the description for a new topic</td>
<td>135</td>
</tr>
<tr>
<td>B.7</td>
<td>Paper &amp; pencil prototype: Notify other users about a new Jotting</td>
<td>136</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

B.8 Paper & pencil prototype: recalling a Jotting .................. 137
C.1 Modified start screen of the BT Digital Library ............... 139
C.2 Initial results display: version clustering topics ............ 140
C.3 Initial results display: version alphabetical order of topics .. 140
C.4 Ontology navigation box with initial result set .......... 141
C.5 Browsing the topic ontology, result for the topic bank automation .. 141
C.6 Displaying the data record for a resource .................. 142
C.7 Ontology navigation box from a data record of a resource .. 143
C.8 Web browser with installed toolbar for Jottings ............ 144
C.9 Creating a Jotting with suggest and relevant topics .......... 145
C.10 Topic description .............................................. 145
C.11 Search for the topic RDF .................................. 146
C.12 Browsing the topic ontology .................................. 146
C.13 Searching for “knowledge description” ..................... 147
C.14 Topic matching “knowledge description”: “knowledge representation” 147
C.15 Selecting a topic from “my topics” .......................... 148
C.16 Creating the new topic “Resource Description Framework” .. 148
C.17 Entering a topic description ................................. 149
C.18 The new topic “Resource Description Framework” .......... 149
C.19 Notifying other users about a Jotting ....................... 150
C.20 Toolbar with activated “edit” and “delete” buttons and active topics 150
C.21 Recalling a stored Jotting ................................. 151
List of Tables

A.1 Summary of information search tactics and definitions. From [20]. . . 127
Appendix A

Additional Resources

A.1 Information Search Tactics by Bates

<table>
<thead>
<tr>
<th>Monitoring Tactics</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1.CHECK</td>
<td>To review the original request and compare it to the current search topic to see that it is the same.</td>
</tr>
<tr>
<td>M2.WEIGH</td>
<td>To make a cost-benefit assessment, at one or more points of the search, of current or anticipated actions.</td>
</tr>
<tr>
<td>M3.PATTERN</td>
<td>To make oneself aware of a search pattern, examine it, and redesign it if not maximally efficient or if out of date.</td>
</tr>
<tr>
<td>M4.CORRECT</td>
<td>To watch for and correct spelling and factual errors in one’s search topic.</td>
</tr>
<tr>
<td>M5.RECORD</td>
<td>To keep track of trails one has followed and of desirable trails not followed up or not completed.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>File Structure Tactics</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1.BIBBLE</td>
<td>To look for a bibliography already prepared, before launching oneself into the effect of preparing one; more generally to check to see if the search work one plans has already been done in a usable form by someone else.</td>
</tr>
<tr>
<td>F2.SELECT</td>
<td>To break complex search queries down into sub-problems and work on one problem at a time.</td>
</tr>
<tr>
<td>F3.SURVEY</td>
<td>To review, at each decision point of the search, the available options before selecting.</td>
</tr>
<tr>
<td>F4.CUT</td>
<td>When selecting among several ways to search a given query, to choose the option that cuts out, eliminates, the largest part of the search domain at once.</td>
</tr>
<tr>
<td>F5.STRETCH</td>
<td>To use a source for other than its intended purposes.</td>
</tr>
</tbody>
</table>
A. Additional Resources

F6.SCAFFOLD  To design an auxiliary, indirect route through the information files and resources to reach the desired information.

F7.CLEAVE  To employ binary searching in locating an item in an ordered file.

Search Formulation Tactics:

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1.SPECIFY</td>
<td>To search on terms that are as specific as the information desired.</td>
</tr>
<tr>
<td>S2.EXHAUST</td>
<td>To include most or all elements of the query in the initial search formulation; to add one or more of the query elements to an already-prepared search formulation.</td>
</tr>
<tr>
<td>S3.REDUCE</td>
<td>To minimize the number of elements of the query in the initial search formulation; to subtract one or more of the query elements from an already-prepared search formulation.</td>
</tr>
<tr>
<td>S4.PARALLEL</td>
<td>To make the search formulation broad (or broader) by including synonyms or otherwise conceptually parallel terms.</td>
</tr>
<tr>
<td>S5.PINPOINT</td>
<td>To make the search formulation precise by minimizing (or reducing) the number of parallel terms, retaining the more perfectly descriptive terms.</td>
</tr>
<tr>
<td>S6.BLOCK</td>
<td>To reject, in the search formulation, items containing or indexed by certain term(s), even if it means losing some document sections of relevance.</td>
</tr>
</tbody>
</table>

Term Tactics:

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1.SUPER</td>
<td>To move upward hierarchically to a broader (superordinate) term.</td>
</tr>
<tr>
<td>T2.SUB</td>
<td>To move downward hierarchically to a more specific (subordinate) term.</td>
</tr>
<tr>
<td>T3.RELATE</td>
<td>To move sideways hierarchically to a coordinate term.</td>
</tr>
<tr>
<td>T4.NEIGHBOR</td>
<td>To seek additional search terms by looking at neighboring terms, whether proximate alphabetically, by subject similarity, or otherwise.</td>
</tr>
<tr>
<td>T5.TRACE</td>
<td>To examine information already found in the search in order to find additional terms to be used in furthering the search.</td>
</tr>
<tr>
<td>T6.VARY</td>
<td>To alter or substitute one’s search terms in any of several ways.</td>
</tr>
<tr>
<td>T7.FIX</td>
<td>To try alternate affixes, whether prefixes, suffixes, or infixes.</td>
</tr>
<tr>
<td>T8.REARRANGE</td>
<td>To reverse or rearrange the words the words in search terms in any or all reasonable orders.</td>
</tr>
<tr>
<td>T9.CONTRARY</td>
<td>To search for the term logically opposite form that describing the desired information.</td>
</tr>
<tr>
<td>T10.RESPELL</td>
<td>To search under a different spelling.</td>
</tr>
</tbody>
</table>
A.2 Five Idea Tactics by Bates

Five of Bates “Idea Tactics”, which can be supported by knowledge management systems to foster the searches, which have reached a dead end. Especially the tactics wander, focus, and dilate can be supported by an application using ontologies.

Consult To ask a colleague for suggestions or information in dealing with a search.

Wander To move among one’s resources, being receptive to alternative sources and new search ideas triggered by the materials that come into view.

Focus To look at the query more narrowly, in one or both of two senses: (1) to move from the whole query to a part of it or (2) to move from a broader to a narrower conceptualization of the query.

Dilate To look at the query more broadly, in one or both of two senses: (1) to move from a part of the query to the whole query or (2) to move from a narrower to a broader conceptualization of the query.

Skip To shift laterally in one’s view of the query, in one or both of two senses: (1) to move from search one part of a complex, multipart query to another one or (1) to view the query form, and search on another angle, that angle being neither broader or narrower, but simply different. [19]
A. Additional Resources

A.3 Rough Data Model

Figure A.1: A rough data model for the use case Jottings
Appendix B

Artifacts of GUI-Design

All models which were used for the GUI-Design are attached to the thesis in this subsection.

B.1 Paper & Pencil Prototype

This prototype was created as an initial effort to design the proposed prototype. It was evaluated with 3 users.
B. Artifacts of GUI-Design

Figure B.1: Paper & pencil prototype: Start screen of the BT Digital Library
Figure B.2: Paper & pencil prototype: Result set of a search
Figure B.3: Paper & pencil prototype: Result record for one resource
Figure B.4: Paper & pencil prototype: Advanced search features
B. Artifacts of GUI-Design

Figure B.5: Paper & pencil prototype: Creating a new Jotting
Figure B.6: Paper & pencil prototype: entering the description for a new topic
Figure B.7: Paper & pencil prototype: Notify other users about a new Jotting
Figure B.8: Paper & pencil prototype: recalling a Jotting
Appendix C

Artifacts of Evaluation

C.1 Prototype of Evaluation: Search & Browse

Figure C.1: Modified start screen of the BT Digital Library
C. Artifacts of Evaluation

Figure C.2: Initial results display: version clustering topics

Figure C.3: Initial results display: version alphabetical order of topics
C. Artifacts of Evaluation

Figure C.4: Ontology navigation box with initial result set

Figure C.5: Browsing the topic ontology, result for the topic bank automation
C. Artifacts of Evaluation

Figure C.6: Displaying the data record for a resource
Figure C.7: Ontology navigation box from a data record of a resource
C. Artifacts of Evaluation

C.2 Prototype of Evaluation: Jottings

Figure C.8: Web browser with installed toolbar for Jottings
C. Artifacts of Evaluation

Figure C.9: Creating a Jotting with suggest and relevant topics

Figure C.10: Topic description
C. Artifacts of Evaluation

Figure C.11: Search for the topic RDF

Figure C.12: Browsing the topic ontology
C. Artifacts of Evaluation

Figure C.13: Searching for “knowledge description”

Figure C.14: Topic matching “knowledge description”: “knowledge representation”
C. Artifacts of Evaluation

Figure C.15: Selecting a topic from “my topics”

Figure C.16: Creating the new topic “Resource Description Framework”
C. Artifacts of Evaluation

Figure C.17: Entering a topic description

Figure C.18: The new topic “Resource Description Framework”
C. Artifacts of Evaluation

Figure C.19: Notifying other users about a Jotting

Figure C.20: Toolbar with activated “edit” and “delete” buttons and active topics
C. Artifacts of Evaluation

Figure C.21: Recalling a stored Jotting

C.3 Questionnaire

Questionnaire for Usability study\(^1\)

Comments before showing the prototype:

- Name
- Advanced bookmarks and Searching and Browsing in a Digital Library
- The application is tested, not the “intelligence or cleverness” of you as a user!
- Think aloud.

After showing prototype:

1. How would you classify your job?

   (a) Managerial
   (b) Engineering (hardware/software)
   (c) Secretarial
   (d) Clerical
   (e) Other

2. Used the digital library before? How long have you been using the digital library?

\(^1\)With ideas from Mayhew [49]
C. Artifacts of Evaluation

3. What do you think about the term "Jotting"? Do you think it describes the functionality?

4. Do you think the searching and browsing is better, the same or worse, as before? Why?

5. Which option of displaying topics did you like more? Cluster or alphabetically ordered?

6. Do you think the Jotting is useful, is not useful? Why?

7. Do you think you are able to classify web pages with topics? Is it helpful?

8. Would you like to create connections/links/relations between topics?

9. Would you like to be able to delete topics again once you have created them?

10. Do you have additional thoughts, comments or critique? Do you miss a function?