

**Knowledge Management based on a
Question Answering System**

**Design and Implementation of a Web Based Integrative
Prototype in Java Struts**

eingereicht von:
Franz Inselkammer

DIPLOMARBEIT

zur Erlangung des akademischen Grades
Magister/Magistra rerum socialium oeconomicarumque
Magister/Magistra der Sozial- und Wirtschaftswissenschaften
(Mag.rer.soc.oec)

**Fakultät für Wirtschaftswissenschaften und Informatik,
Universität Wien**
**Fakultät für Technische Naturwissenschaften und Informatik,
Technische Universität Wien**

Studienrichtung: **Wirtschaftsinformatik**

Begutachter:

Univ.-Prof Dr. Stefan Biffli
Dr. Alexander Schatten

Wien, 5.3.2004

Abstract

This paper will particularly describe a new approach of knowledge management based on a prototype of a question answering system, which was created belonging to that diploma thesis. The paper will precisely describe the advantages and disadvantages of some existing systems. A part from that we will explain the basic concepts of knowledge management on reference to the prototype. Another important aspect is the integrative effect of the new system which makes it possible to include different resources in one coherent system. Based on that we elucidate the process of stemming documents to make them avail- and usable for the knowledge database. Finally the software engineering process of the prototype will be illustrated.

TABLE OF CONTENTS

1	Introduction to QAS-Systems	6
1.1	Introduction.....	6
1.2	How to Design a Knowledge Management System	6
1.3	Natural Question Answering	6
1.3.1	The Problems about QAS	7
1.3.2	Users	7
1.3.3	Question.....	7
1.3.4	Answers	7
1.3.5	Presentation.....	8
1.3.6	History of QAS	8
1.3.7	BASEBALL.....	8
1.3.8	Evaluation of a Question Answering System.....	9
1.3.9	LUNAR.....	10
1.3.10	Interpretation of Question.....	11
1.4	AnswerBus - a Web Based Question Answering System	12
1.4.1	AnswerBus - a Multilingual System.....	12
1.4.2	The Methodology of AnswerBus.....	13
1.4.3	Conclusion.....	15
1.5	Cyc - The Approach to Manage Common Knowledge	16
1.5.1	Open Source.....	16
1.5.2	Use as QAS	17
1.5.3	Cyc in Use.....	17
2	Knowledge Management	19
2.1	The Beginning of Knowledge Management	19
2.2	Basic of Knowledge Management	20
2.2.1	Terms and Development	20
2.2.2	The Difference between Information and Knowledge	22
2.3	Definition of Knowledge Management	23
2.4	Aspects of Knowledge Management	23
2.4.1	Introduction.....	23
2.4.2	Informatics and Knowledge Management	24
2.4.3	Business Practice and Knowledge Management	24
2.4.4	Strategy Research and Knowledge Management	25
2.5	Challenge of Knowledge Management Systems	25
2.5.1	Challenge of Knowledge	25
2.5.2	Knowledge as Resource	26
2.5.3	Changing Established Habits	26
2.5.4	Communication of Knowledge	27
2.5.5	The Management of Knowledge	28
2.5.6	Knowledge Management	30
3	Information Retrieval and Processing	31
3.1	Stemming	31
3.1.1	Introduction.....	31

3.1.2	Stemming and Decomposing	31
3.1.3	Stemming German compared to English.....	32
3.2	Stemming German.....	33
3.2.1	Compounds	33
3.2.2	Characteristics of the German Language	34
3.2.3	Stemming and Decomposing Approaches	34
3.2.4	Stemming German Texts	36
3.2.5	Discriminating, Substitution and Stripping of German Text	37
3.2.6	Character Substitution.....	38
3.2.7	Suffix-Stripping	38
3.2.8	Evaluation.....	39
3.2.9	Understemming.....	39
3.2.10	Overstemming Nouns	40
3.2.11	Improvements	41
3.3	Stemming The English Language.....	42
3.3.1	Introduction to the Porter Stemming Algorithm.....	42
4	Managing the Information Flood	43
4.1	Finding and Preparing Information	43
4.2	Weak Points in Detecting Information.....	44
4.3	Quality of Information.....	44
4.4	Structure of Data.....	45
4.5	Usability	45
4.6	Metadata	45
4.7	Access Control.....	45
4.8	Multilanguage	45
4.9	Filtering Information.....	45
4.10	Improvements by the Prototype	46
4.11	Conclusion.....	47
5	Description of the Implemented Prototype	48
5.1	Description of the User Interaction Scenario	48
5.1.1	Introduction.....	48
5.1.2	Step 1: Answer given directly by Knowledge Database	48
5.1.3	Step 2: Answer given by Local Information Resource.....	49
5.1.4	Step 3: Answer given by a Global (External) Resource	50
5.1.5	Step 4: Answer given by Trusted User(s).....	52
5.1.6	Step 5: Management Activities.....	53
5.1.7	The Scoring System	54
5.1.8	Sequence Diagram.....	56
5.1.9	System Integration – Technical Aspects.....	57
5.1.10	Motivation and Cost Saving Factors.....	58
5.1.11	Access Control.....	59
5.2	Introduction to Lucene.....	59
5.3	Introduction to the Prototype of the Question Answering System	65
5.3.1	The Question-Based Approach.....	65
5.3.2	The User Interface	66
5.3.3	Indexing and Searching in Detail	71

5.4	Class Diagram.....	72
5.5	Scenario for a Test	73
5.5.1	Testing the Usability.....	73
5.5.2	Testing the Code.....	74
5.6	The scientific discoveries and benefits of that paper	75
5.7	10 Theses and Future Outlook	76
5.8	Facts about the Prototype.....	77

1

Introduction to QAS-Systems

1.1 Introduction

To have an idea what a knowledge management system based on a question answering system is about we have a look at different question answering systems in this chapter. Based on that knowledge we can go on in Chapter two and explain the principles of knowledge management. Chapter three tries to connect those components and gives a closer look at some important techniques which are necessary to convert the theory into praxis. Chapter four finally describes the implementation process of the prototype being developed parallel to that paper. I describe a design process, which leads to the development of a knowledge management system. The prototype of that work will be a knowledge management system which gathers its knowledge by collecting answers and questions from users. Helpful resources to find suitable answers are files, databases, search engines or other users who participate in the system.

1.2 How to Design a Knowledge Management System

Knowledge Management Software must be embedded in processes of knowledge workers' everyday practice. In order to attain a proper design, regarding the special qualities and requirements of knowledge work, participation of the knowledge owners and future users is an important factor for success of knowledge management systems. But the reality is different. In order to bring organizations closer to their expectations and visions, embedding knowledge management in everyday work is the most important. In this chapter I will describe characteristics of knowledge work motivating the usage of participatory design techniques. I introduce a design process for developing and improving knowledge management, which includes ethnographic surveys¹ and usage improvement by time.

1.3 Natural Question Answering

As the richness of information in the World Wide Web grows and users get used to the wealth of information the need for a automated answering system becomes more urgent. We need a system that allows a user to ask questions in a specific language and receive an satisfactory answer quickly. In addition the system has to validate if the given answer matches the requirements of the user. The problem of current search engines is that they return ranked lists, but do not give answers to the user.

¹ ethnographic surveys in that case means that users sharing the same problems can incorporate and find each other

Question answering Systems treat with that problem. By now good systems are able to answer two third of the asked factual questions. The combination of user demand and the promised results have stimulated the effort and success of Question-Answering-Systems.

1.3.1 The Problems about QAS

Firstly to answer a question the system has to analyse the question. On possibility to do that is by considering the actual business context, another by consulting the local database system. The System has to find one or more answers by contacting all kind of resources, according that the QAS has to provide an answer to the user in an appropriate form possibly with multimedia information.

We can subdivide QAS according to the source of the answers. The source can be for example a single data base or structured data, semi-structured data like comment fields in data or free text like available in the internet. Further we distinguish among search over a fixed set of collections or the search over a collection or book like an encyclopaedia. In help systems we can distinguish between domain independent and domain specific QAS.

1.3.2 Users

We distinguish different types of users. The system has to be designed in a way that both first time or casual users as well as “power users” are able to such a system. These users need different functionality, ask different questions and require another kind of answers.

1.3.3 Question

Generally questions are distinguished by their answers. Answers can be factual answers, opinions or summaries. Generally it is difficult to differ between these three types of answers. Furthermore the type of question makes no difference to the answer since the comparison and the storage of a question and answer set is the same. Next we detect different kind of questions like questions which can be answered with “yes or no”, or the so-called “wh” questions. These are questions which either begin with “*who* is the president” or with “*how* much is your weigh”. There is an evidence that *why* and *how* questions tend to be more difficult to answer, because they require understanding causality or instrumental relations and these are typically expressed as clauses or separate sentences.

1.3.4 Answers

Answers can be short or long, a list, a summary or just a diagram or a picture. Answers differ, depending on who will be the receiver and who will be the sender. Two users who are used to work together a while may have developed a language of shortcuts which could be incomprehensible for other users. There are also different methodologies for constructing an answer. It is possible to give an exact answer or just to extract snippets from a document. The second solution is utilized by many search engines in the internet. If the answer is drawn from multiple sentences or various documents you have to take into account that the interconnection of the answer is low and maybe obscure. In that case the user has to trace out the necessary parts of the answer which is relevant for him.

The question is what makes an answer satisfactorily! Has an answer to be short or long? The point is if an answer is derived from an external resource which was generated automatically the system should present multiple answers. This allows the user to find a correct answer out of some available ones or out of a whole document. The hit ratio for

an exact and correct answer given by an automatically generated answering system is due to the complexity very low. On the other hand an answer given from a natural person should, if possible, be short and precise.

1.3.5 Presentation

Normally a user asks a question and receives one or many answers from another user, of the system or from external resource. If there are too many answers from an external resource the user tries to specify his question to receive a more meaningful set of answers. Facilitating such dialogs would help both usability and user satisfaction. Thinkable would be a system, which has the ability of natural voice recognition. Such a question answering system could provide conversational access to the internet and the information in the web. This area could be of great commercial interest for telecommunication and web content providers. Think about a 24 hour telephone help line, which is able to answer frequently asked questions. When we consider current technologies and their fast developments this idea could be reality in a couple of years. So far there has been little work on interfaces for question answering systems. There have been a few systematic evaluations on how to present the answers and information best to the user and how much context and how many answers to satisfy the users needs [Hirschman 1998]. This is an area which will begin to grow in the next years as commercial question answering interfaces begin to be deployed.

1.3.6 History of QAS

The interest in Question Answering Systems grew in the past years enormously. Especially systems based on natural question answering became attention since the introduction of the Question answering track in the text Retrieval Conference which began with Trec-8 in 1999 [Mayfield 1999]. This conference was not the first time that the topic has been addressed by natural language processing researchers. In fact Simmons [Simmons 1965] begins a survey article "Answering English Questions by Computer". The publication of his paper led to the fact that about fifteen implementations of question answering systems were built in the preceding five years. This systems include conversational question answers, front end to structured data repositories and systems which try to find answers to questions from text sources, such as encyclopaedias.

1.3.7 BASEBALL

One of the first question answering systems was a program called BASEBALL [Baseball 1961]. As one can guess, it is a program about Baseball games and the league in America. The system was able to answer questions like "*who did the Red Sox win on August the fifth?*" or "*How many games did the Yankees play in October?*" The system was even able to answer complex questions like "*On how many days in July did eight teams play?*". BASEBALL analysed the question, using linguistic knowledge, and put it into a canonical form which was then used to generate a query against the structured database containing the baseball data. BASEBALL was at this time a very sophisticated program, even by current standards. The way it dealt with syntax and semantic was outstanding at that time. The whole system had the drawback that it was restricted to one domain, namely baseball.

In addition BASEBALL has been designed as an interface to a database and not as an interface to a collection of incoherent documents. This made it impossible for the system to use it as an Question Answering System which is based on the content of the World Wide Web. In this regard BASEBALL was the first of a series of programs designed as a “natural language front-ends to databases”. The assumption at this time was, that databases hold vast amounts of structured data. The details of that databases would be opaque to many users. Rather than compel time-pressured users or computationally-challenged executives, to learn the structure of the database and a the specialised language for querying it, the aim was to allow users to communicate in their own language by using an interface which knows about questions and the database structure.

1.3.8 Evaluation of a Question Answering System

To evaluate a question answering like BASEBALL or the prototype, which will be implemented parallel to this paper, we try to consider some selected factors, which seem to be important for a modern system as well as for the prototype. As there is no standard on how to evaluate question answering knowledge management systems or only answering systems I developed a procedure how to measure the quality of such a system. One has to take into account some sometimes real results of systems are not available and for that can only be taken from the literature. The factors we considered for the evaluation are conditions we want to meet perfectly for the prototype. The evaluation is achieved by six factors. The *ability to expand* is an assessment to show if the system has the possibility to be expandable. This could be some kind of plug-in to expand the domain or other functionalities like add-ons or simply the possibility to expand the program. The more expandabilities the system has the better it is, which means the bigger the filled red area. BASEBALLS’ expandability is very low, which is shown in a small red area over the criterion.

Ability to learn is a criterion which signifies if a system is able to learn and to improve its answers over time. A good system should be able to learn automatically from the user. BASEBALL for instance can only improve its answers if a programmer improves the analysing process or if the database is fed with more rules and information. It does not have an interactive learning process by giving a correct answer which will be confirmed by the user as well as a wrong answer which will be refused by the user.

The *natural language input* should be one of the major criterions of an modern question answering systems. Users with only a little computer knowledge should not be forced to learn a specific query language. Computers should have the ability to interpret natural language and to translate it to their language. We can not expect that every user is familiar with computers. BASEBALL understands natural language input, but is limited to its domain, which of cause is baseball and its environment. A question about the weather would properly be misinterpreted.

The factor *no domain restriction* signifies if a system is restricted to one domain. On the other hand it is obvious, that systems which are restricted to one domain, do very often have a outstanding quality of their answers, if the question belongs to that domain, too. BASEBALL is restricted to one domain which has an negative effect to our assessment.

Multilingual is a criterion to determine if a system supports more than one language. The more languages the system supports the better it is. The BASEBALL system only supports English.

As already mentioned before, another important factor is *the quality of the answer*. The quality here can be seen as preciseness of the answer. BASEBALL for instance returns only one answer which is exact and straight forward. Other systems may return a set of answers whereby the user has to fish out his matching one.

The following graphic will be shown at the end of every question answering system which will be analysed in this paper. The bigger the red area the better the system.

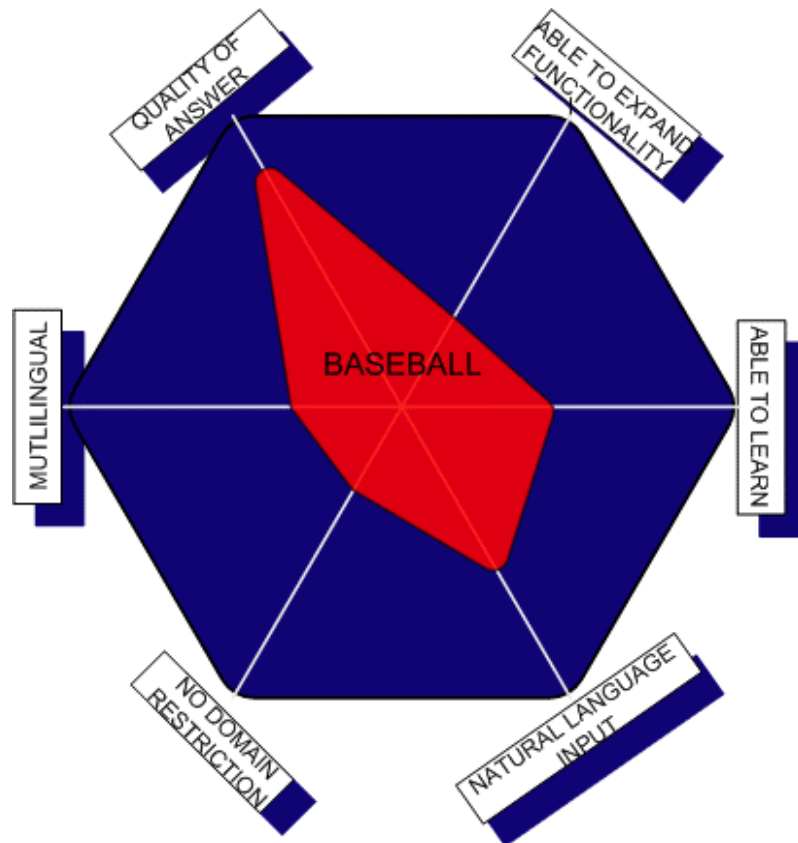


Image 1.1 Baseball Evaluation

1.3.9 LUNAR

Another well known work in this field is a system called LUNAR [Copestake 1990]. This program was designed to enable a lunar geologist to comfortably access, compare and evaluate the chemical analysis data on lunar rock and soil composition that was collected as a result of the Apollo moon mission. LUNAR was able to answer questions like *What is the average concentration of aluminium in high alkali rocks?* or *How many Breccias contain Olivine?* The system was presented at a lunar science convention in 1971 and was able to answer 90% of the questions, which of course were in the field of lunar rocks, posed by the geologists. Even the phrasing was arbitrary and no instructions were given on how to ask questions. But again one has to take into account, that the Question Answering System is an in-domain system, which cannot be carried out to a common solution.

In our system the primary task is to create a in-domain interface. Later on in a further developed status we can concentrate on an external interface which enables a user to find an answer to an in-domain question on the web or by means of other resources.

From the perspective of the current research focus in question answering, the key limitation of that work is, that it presumes the knowledge the system is using to answer the question is a structured knowledgebase in a limited domain, and not an open ended collection of unstructured texts is used by the system to answer questions. The maintenance and the creation of the data structure itself is a major part of the system challenge.

From our point of evaluation the system is very similar to the BASEBALL system, so it is no wonder that the following evaluation graphic is close the previous one.

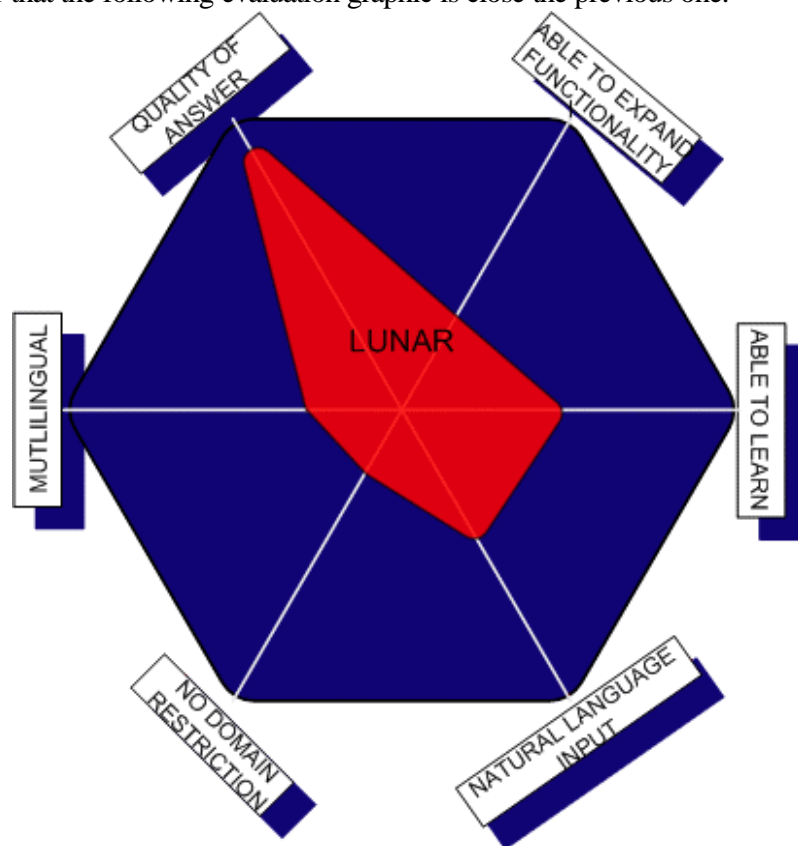


Image 1.2 Evaluation Lunar

1.3.10 Interpretation of Question

Another important factor is the interpretation of a question. For example if the answer to the question *Do you know the time?* is *Yes!* or the answer to *How did John pass the exam?* is *with a pen!* the user might be less than satisfied. For that reason Wendy Lehnert introduced thirteen conceptual categories of questions, such as “Verification”, “Request”, “Causal Antecedent”, “Enablement”, “Instrumental or Procedural” etc. In 1977 she introduced a system called QUALM [Lehnert 1997]. Her key concern in this work was to move away from the view that natural language question answering should be seen merely as a front-end to a completely separate data or information retrieval process.

Instead she viewed the process of question answering as one in which both the understanding and answering of a question relies on the context of the story and pragmatic notions of appropriateness of answer.

Another point about that is the idea of a recommendation engine. Sometimes it might help other users to see what kind of questions have been asked in correlation to others or straight after another. A user A who asks a question to which she received an unsatisfactory answer might ask another questions to which she received an satisfactory answer. The two questions are correlating somehow. Another user B asking question one could now be offered the second question of user A as a substitute. The conclusion could be that this correlations might help to come closer to the desired answer. For example User A wants to know *who was the president in 1977?*, but he does not receive an answer to that question. There upon he asks *who was the leading party in 1977?* and receives an answer. Anyone in the future asking for the president in 1977 should easily have access to the question of the leading party in 1977, too.

1.4 AnswerBus - a Web Based Question Answering System

1.4.1 AnswerBus - a Multilingual System

AnswerBus [Answerbus] is an open-domain question answering system based on sentence level information retrieval. It accepts users' natural-language questions in English, German, French, Spanish, Italian and Portuguese and extracts possible answers from the Web. This makes the project interesting for our work, as our prototype will use a similar technology. AnswerBus can respond to users' questions within several seconds. Five search engines and directories (Google, Yahoo, WiseNut, AltaVista, and Yahoo News) are used to retrieve Web pages that potentially contain answers. From the Web pages, AnswerBus extracts sentences that are determined to contain answers. The current rate of correct answers is good.

AnswerBus should demonstrate that practical question answering on the Web is feasible.

AnswerBus takes a user question in natural language. A simple language recognition module will determine whether the question is in English, or any of the other five languages. If the question language is not English, AnswerBus will send the original question and language information about the question to AltaVista's translation tool BabelFish [Babelfish], and obtain the question that has been translated into English. A practical test where I tried to answer the question "When was J.F. Kennedy killed" in English and German

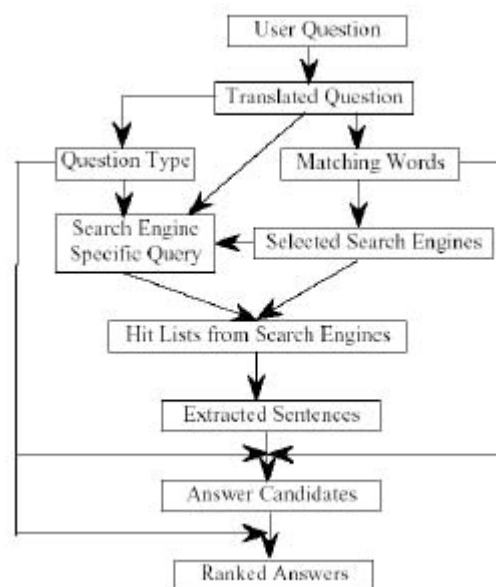


Image 1.3: AnswerBus

did not return the same result. While the English version of the question returned some suitable answers, the German question “Wann wurde J.F. Kennedy getötet” brought no answer at all. This concludes that the translation process needs some improvement. The following graphic shows an overview on how AnswerBus works.

The rest of the process consist of four steps:

1. select two or three search engines among five for information retrieval and form search engine specific queries based on the question
2. contact the search engines and retrieve documents referred at the top of the hit lists
3. extract sentences that potentially contain answers from the documents
4. rank the answers and return the top choices with contextual URL links to the user. Instead of returning a snippet of fixed length of text, AnswerBus returns sentences, which can provide users with some contextual information for the answers.

AnswerBus aims to retrieve enough relevant documents from search engines within a response time that is acceptable to users. The main tasks are to select one or more appropriate search engines for a specific user question, and then form queries that match to the question as well as the selected search engines. The formation of the queries is an essential procedure because it can largely influence the recall and accuracy of question answering and the speed of the system operation.

Different search engines or directories may match different types of questions. For example, for questions about current events, Yahoo News may be a better choice than Google. For a specific question, AnswerBus chooses to use two to three most appropriate search engines among the five ones. In order to determine which search engines are best suited for a specific question, AnswerBus preanswered 2000 sample questions, including questions typed in by test users. It sent the queries based on these questions to all of the five search engines. For each question, it recorded the number of possible answers that came from the different search engines. All the words used in queries are indexed. For example, for Word 1, Google may return 8 answers, AltaVista returns 4 answers, and Yahoo returns 7 answers; for Word 2, Google 6 answers, AltaVista 6 answers and Yahoo 5 answers. For a query with Word 1 and Word 2, AnswerBus will choose to use Google (8+6) and Yahoo (7+5). If a query contains words not included in the indexed list, AnswerBus uses the search engines’ average returns for all the indexed words to determine which search engines are most appropriate.

1.4.2 The Methodology of AnswerBus

Most search engines are not designed for natural language questions. The computing speed has been regarded as very important throughout the development of AnswerBus. For a scalable Web-based QA system, compromises need to be made between speed and recall of answers. Thus AnswerBus does not try to find the best query; instead, it tries to locate the good enough query that will conduct the search task very fast. The focus has been laid on generating one simple query instead of an expanded one.

Several approaches are combined to form queries, including functional words deletion, use of word frequency table, special words deletion, and word form modification.

Especially these approaches are of interest for further development of our question answering prototype.

- *Functional words deletion*
Functional words include prepositions, determiners or pronouns, conjunctions, interjections, and discourse particles. Functional words deletion, which often can make the query short enough, can be used as a baseline of a search engine specific query formation. Some words are not functional words but are acting as functional words for example, “kind of,” “name the designer of” can also be deleted from the query:
- *Use of word frequency table:*
Another way to make a query shorter is to delete frequently used words in the query. The basic idea is that the more frequently a word is used in a language, the less discriminating the word is. Thus AnswerBus implements a word frequency table. For a long query, AnswerBus sorts all the words in the query and deletes one or more words that are identified as frequently used by the frequency table.
- *Word form modification*
Some words in the original question are converted to another form then put in the query. Usually they are verbs, for example, *ended* becomes *end* in questions like “When did the Jurassic Period end?”. *Has* becomes *have* in questions like “How many hearts does an octopus have?”

After sending the question to the preferred SearchEngines, AnswerBus downloads and processes the documents referred at the top of search results returned by different search engines. First of all it parses the documents into sentences and then determines whether a sentence is an answer candidate through a process of word matching. Answerbus uses a HTML tool to split of the HTML tags and to receive the intrinsic text.

In order to determine whether a retrieved sentence is potentially an answer to the question, AnswerBus classifies all words in the original question or sentences in the retrieved documents into two categories: matching words and non-matching words. All words that are used to form the search engine specific query are matching words. The rest are non-matching words.

After the extraction of answer candidate sentences, each sentence has received a primary score. Those sentences with a score of “0” are dropped. Nevertheless, the primary scores are not robust enough for the judgment whether a sentence is a real answer. AnswerBus uses several techniques to refine the primary scores, which are the determination of question type, the use of a Question Answering specific dictionary and named entities extraction. The final score that is used to determine the rank of an answer is a combination of the primary score and the influence of all the different factors.

Almost all question answering systems use question type to judge the answer. They classify the question types in different categories, or on that, what users expect to receive. For example, a “*Who is ...?*” will be assigned as the type person or organization ; while “*When did ... happen?*” will be classified as a date or time question.

AnswerBus also uses question type as an important piece of information to judge whether a sentence can be an answer to a question. AnswerBus classifies questions into different question types together with some parameters. For example, AnswerBus classifies both of “*How far ...?*” and “*How close ...?*” questions as distance questions, but it also differentiates these two types of questions. The unit of the answer to “*How far*

...?” most likely will be “mile,” “kilometre” and other related bigger units, it has small chance to be “inch,” “centimeter” etc. For “*How close ...?*”. questions, the unit of the answer to this question could be any of the above, depending on the context in the question. It could also be “nanometer” or others. AnswerBus uses a question answering specific dictionary, a database containing this kind of information about the relationship of words between questions and answers. For example, for the entry of word “far,” the definition provided in the dictionary contains “miles,” “kilometers,” “light years;” for the word of “high,” the definition contains “feet,” “meters.” The dictionary is used to distinguish question types, and determine whether a sentence is the right answer.

1.4.3 Conclusion

AnswerBus is a very powerful question answering system. It demonstrates that question answering in the web is not a fiction anymore. The fast response time and the acceptable answers, as long as questions are being asked in English, make the system interesting as an interface add-on for the prototype question answering system which will be developed parallel to that paper. A further description of the prototype’s interfaces and their modules will be described later. The following graphic shows that AnswerBus reaches a high rank considering the factors of evaluation.

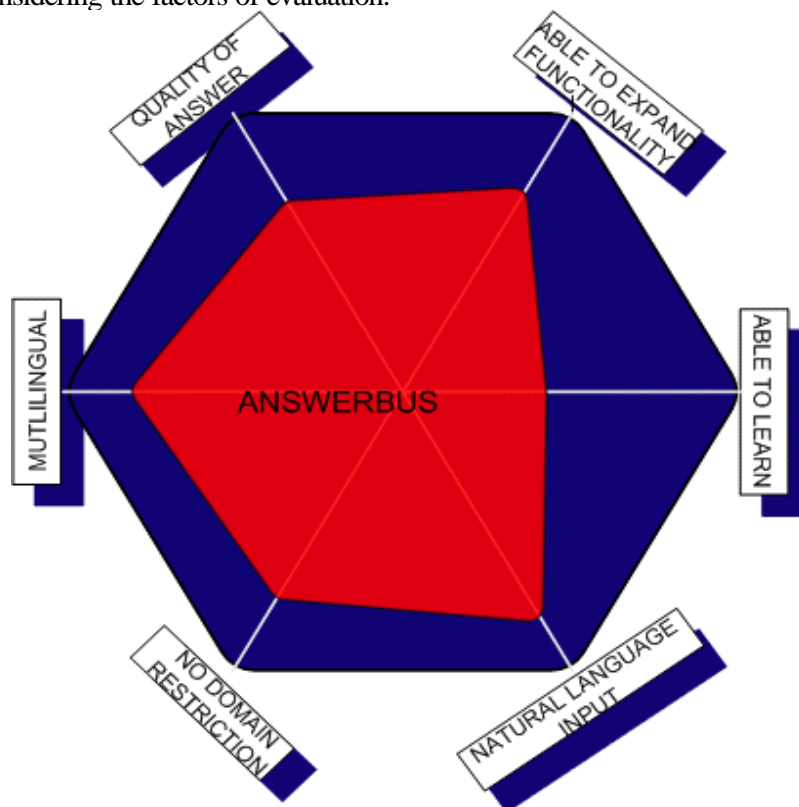


Image 1.4: Evaluation AnswerBus

1.5 Cyc - The Approach to Manage Common Knowledge

In 1984 the American Industry, or to be more precise a research fond to which belong firms like Boeing, HP and Microsoft in Austin, created a project called cyc [cyc], derived from “encyclopaedia”, the answer to the Japanese program of the 5th Computer generation. The reason why we understand clauses like *When I came home a mouse nibbled at my new mouse* is, because we understand the same words with a different meaning. And again the reason why we understand the same words with a different meaning is, because we have a lot of (common-) knowledge about the things that occurred in the sentence above. For example we know, that mice like to nibble. Furthermore you can guess that someone prefers to buy a input device rather than a pet. Exactly that knowledge, or from another point of view, the lack of knowledge is the reason why current computer-based question answering systems and text comprehension systems don’t achieve the breakthrough.

Common knowledge is hard to manage in a system that is based on zeros and ones. Until then it was extremely complicated for both the artificial intelligence and the field of informatics to deal with the implicitness of our culture. Because of that reason employees of Cycorp had to analyse and collect statements with their common sense for ten years now. They systematize statements like *a ticket for a cinema costs 5\$* or *humans live during a single continuous ending time interval*. This could be an important step towards mechanical translation.

Cycorp claims to be "the leading supplier of formalized common sense". CEO and founder Doug Lenat has labored 17 years to codify facts such as "Once people die, they stop buying things." He uses a form of symbolic logic to classify and show the properties of information in a standard way.

CycOrg has put in 600 person-years of effort, and has assembled a knowledge base containing 3 million rules of thumb that the average person knows about the world, plus about 300,000 terms or concepts.

Cyc is more seen as a power or common knowledge source rather than a single application. It could be used for any given application. Sometimes one needs common-sense knowledge and sometimes domain knowledge. Cyc could be used for applications which need common-sense knowledge.

1.5.1 Open Source

To achieve better popularity the organisation created an OpenSource project [opencyc], which allows many people to use the cyc knowledge base and the cyc engine with little restriction compared to the commercial version. The stroke of genius behind the idea is, that the whole community is using the same knowledge base. This of course produces a lot of low quality knowledge or even invalid statements. For that reason CycCorp filters statements if they fit into the general valid knowledge base. If a rule claims that a car makes 500 km/h Cyc will be sceptical against this as it knows that vehicles on earth won’t make more than 280 km/h and sorts this kind of rule out.

1.5.2 Use as QAS

The recognition of coherences and especially the ability to manage common knowledge makes Cyc interesting as a usable technology for an Question Answering System combined with a knowledge base, especially because Cyc is OpenSource and for that free of charge. But the system has two major drawbacks. The first is the fact, that knowledge does not grow by it's own. Every new rule or axiom has to be entered manually, a process that absorbs a lot of patience and time. Furthermore an employee is needed, who is familiar with the CycL programming language and the way how to enter knowledge rules. The second drawback is the complexity of Cyc. It would need month to install and implement a system that is based on the knowledge base of Cyc. Time and money which is not available for our QAS at present time.

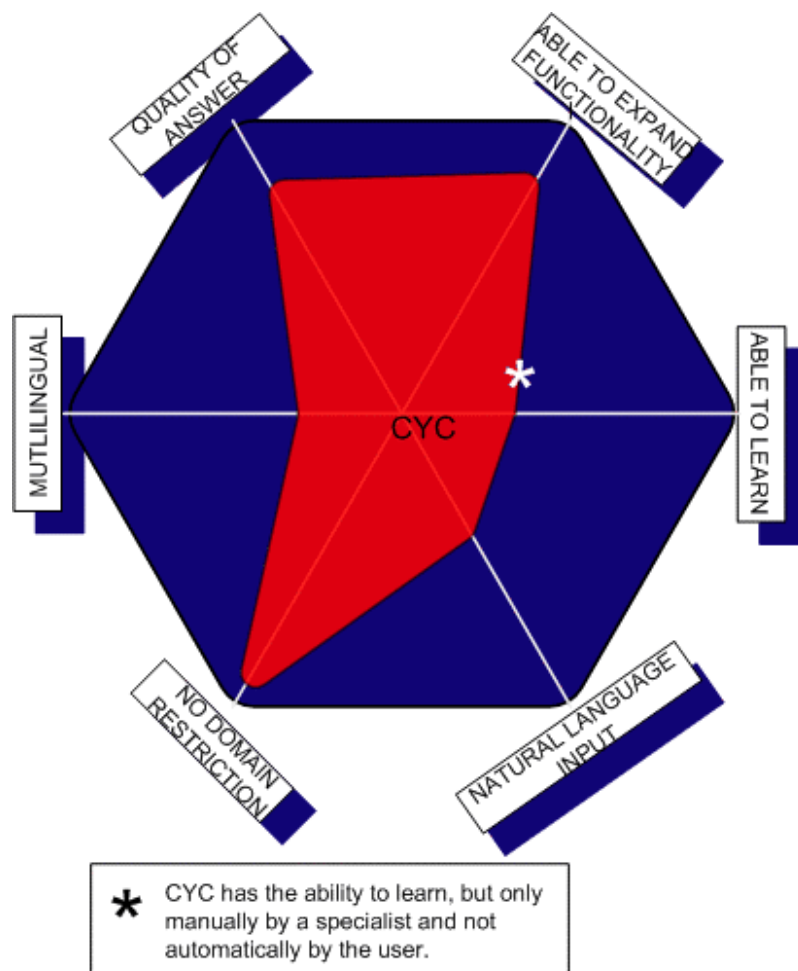


Image 1.5: Evaluation Cyc

1.5.3 Cyc in Use

Ten years ago the founder of Cycorp Doug Lenat prefigured that Cyc would be in every Microsoft Operating System. By that time Cyc was a kind of outstanding, but the expectations have not been fulfilled completely. Still Cyc is an interesting product which should not be underestimated.

At present time Cyc is used at a top-secret military command, which is pickier than most organizations about information security. And it is one of the first users of CycSecure, the first commercial application of Cycorp Inc.'s Cyc Knowledge Base. The command uses CycSecure to ensure that all relevant patches have been applied to fix known vulnerabilities in its networked computers.

CycSecure fits into a variety of projects, including the U.S. Department of Defense's Information Assurance Vulnerability Assessment notification program, to keep up to date on all known ways a system can be attacked. It also knows about the military command's computers and networks and combines that knowledge with the vulnerability information to simulate network attacks. When it spots a potential vulnerability in a computer, it can go out to that box to determine whether it is in fact vulnerable and then recommend the appropriate patch.

CycSecure for the command is customized, but because such customization involves just adding rules and knowledge to the database, it doesn't require software changes. Cycorp maintains the application for the command now, but eventually the user will be able to take over maintenance itself.

It's rule-driven; it's dynamic, it kind of grows and it keeps up with the attacks.

Cycorp CEO Doug Lenat offers this explanation of CycSecure: "Cycorp Cyc knows what are normal, legitimate actions -- such as a user renaming one of their own files or changing their password -- and what are actions taken by hackers -- such as packet-sniffing and spoofing. An attack plan generally includes a large number of 'normal' steps and one or more 'hacker' steps. Cyc does not have a model of the hacker mentality, such as goals, ego and so on, but it does have the notion that hackers generally want to be undetected, since that motivation accounts for many steps in many plans which would otherwise be missed."

Knowledge Management

2.1 The Beginning of Knowledge Management

During the time of the last century one could observe how the economy changed. At the beginning of the 20th century the persons in work were made up of 40% of workers in the agriculture, 30% were employed in the industrial sector and about 10% in the service sector. Only about 10% at these days were employed in the knowledge and information sector. While observing the scenario today we notice a total change. The persons who are in the information and knowledge sector have a percentage of over 60% while on the other hand workers in the agriculture sector decreased to 4%. The question is how could a manufacturing sector decrease dramatically while another sector grew by 600%, although the sector “only” produces soft goods which cannot be touched.

From the development of different enterprises we can see that they either survived by expanding or by specialisation. Moreover the last century, many inventions in the agriculture and industry were made so that the need for physical work became less important. In former days many workers were needed to farm a farmstead, while by now the work process can be done by two persons with the help of machines. This was one reason why persons tried to get employed in the industrial sector. Education was not needed in their new jobs as most employees had to do monotonous tasks. Only until the products became more complex and producers noticed that they would need more educated and specialised staff, companies began to spend money for better education and advanced training. Behind that was a problem that was not seen at the beginning. After the educated employees or as we call them now the knowledge owners left the company their specialised knowledge went away, too. All their experience left and if the person had a key position this could be an enormous loss for the company. In some seldom cases it could even endanger its existence. A task which was processed by a worker who retired was now impossible to handle.

While companies were trying to compensate the risk by investing in new technologies they noticed that this alone was not sufficient. Knowledge Management needs a more “holistic” approach. To treat this problem knowledge can be understood as a new resource. Hence, in our days companies try to split the knowledge to many employees to avoid that one person is the owner of the whole business needs. Moreover if a single person has the whole knowledge, most of the time this person is overworked and builds the bottleneck as business procedures can't be done without his help.

The splitting of knowledge and the introduction of knowledge strategy is not as easy as it sounds as knowledge is not a hard good which can be apportioned arbitrarily. Furthermore knowledge can not be measured or copied. One could describe knowledge as a kind of unit which everybody needs, but nobody exactly knows what it costs or how much of it

can be bought for a certain amount of money. It is difficult to say that a trainee program that costs 10€ is worth the same amount or increases the value of knowledge for the individual in the same amount. Too many factors are involved. For instance the individual ability to learn and the existing knowledge in that area. The next point is that someone could say that she learned something but that does not implicitly guarantee that she knows how to apply the new knowledge. It turns out that knowledge is not just everything we see or hear. We always have to group and connect the learned material to already existing knowledge. Only if that happens new knowledge has been created.

2.2 Basic of Knowledge Management

2.2.1 Terms and Development

To introduce the concepts of knowledge management we should have a look at how humans try to understand complex innovations. When humans try to understand non-transparent-facts they mostly use abstraction and differentiation so that the facts are understandable for everybody. The history of knowledge management was determined by practical and theoretical points of views. The theoretical approach was defined by many debates about knowledge while the practical approach has been driven by the question on how to treat the resource knowledge. In between those two we can only find the discussion how the deployment of instruments with which we can handle knowledge. This discussion is based on information technology and knowledge. On the one hand information technology is used in almost every enterprise which qualifies IT as a natural medium to handle the flow of knowledge. The main aims of knowledge management in enterprises or the introduction of an IT-system. On the other hand do knowledge management theoreticians warn of the high cost which do not implicitly lead to a better output. The high investment in new IT-systems can cause a worse output when saving will be made in the employees education instead. The big risk of that strategy is that knowledge becomes a inert information. This way of knowledge management which sometimes is called : “more IT, less human !” can lead into a dead end [Borghoff 1997]

The following graphic should demonstrate how knowledge arises [Probst 1997]

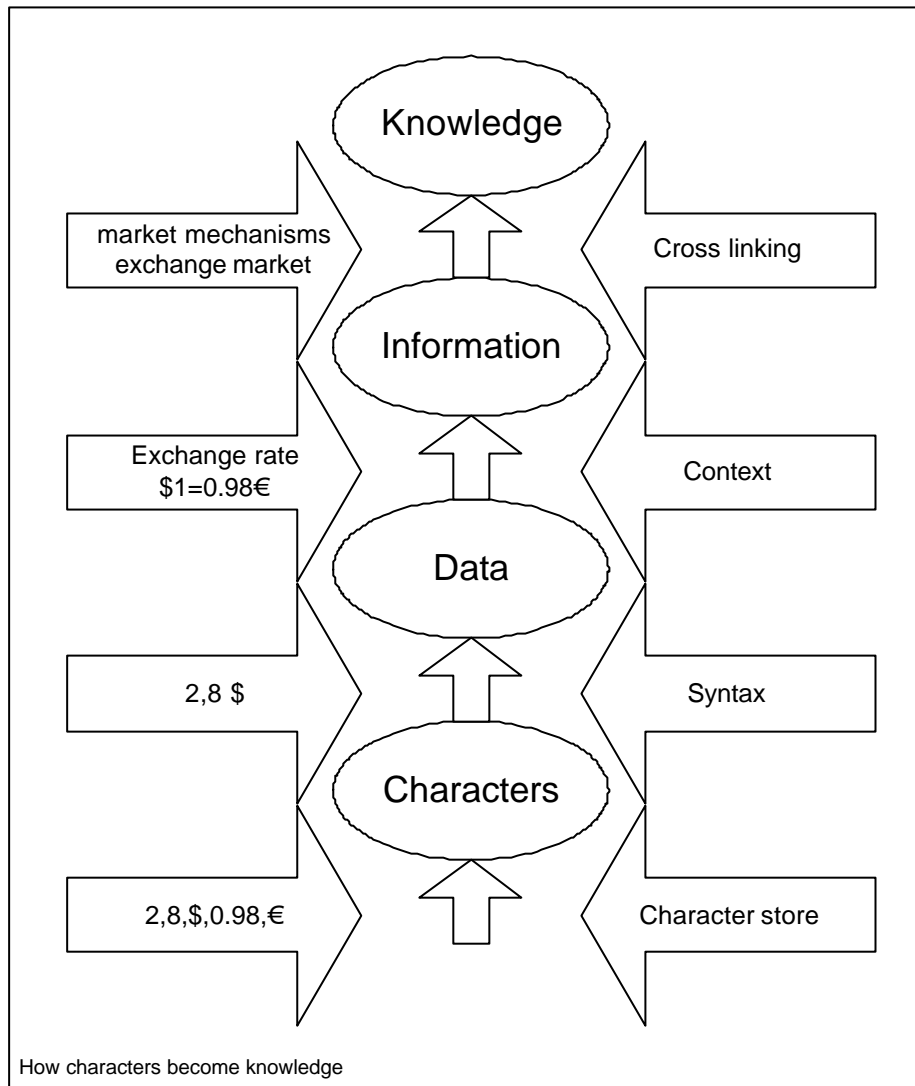


Image 2.1 How characters become knowledge

At the beginning there is an unordered set of characters. Those characters build the presupposition for the following steps and can contain letters, numbers or special characters. These characters are brought into a syntax which makes them readable for both computers and humans – we then talk about data. Data always arises through the combination of characters. If data has a certain context it builds information. If we consider the graphic again we can conclude that information is a combination of data which has a certain context. For that information is always oriented to the receiver and is subject to its individual perception. A receiver does not compulsory have to be a human being. Information can be understood or interpreted by computers as well. The most difficult part now is to find different cross links between information. Most software products fail on that step, because the complexity of it is very high. A nice approach to cross linking knowledge was discussed before when we described the Cyc [cyc] project. Another abstraction nor shown in the graphic could be the competence. It is the ability of an individual to use his knowledge on a certain task, but also the ability to combine

different information in such a way that new knowledge arises. We can say that knowledge is the premise for competence and the basis of acting of an individual.

What knowledge is and how it arises is one of the basic questions of philosophy. Grant [Grant 1996] gives a definition on what knowledge is. This definition is always subjective and depends on the point of view and the area of research one is situated.

“Knowledge is always fixed to memory and arises through the processing of perceived information in our brain. A requirement for the processing in our brain is that contexts, which have been important for the development in the history of the system, are available. The association of perceived relevant information with available contexts or experiences are building the end product of the learning process, in which data is being registered as information and is being learned as new knowledge.”

2.2.2 The Difference between Information and Knowledge

The intent with KM is to manage knowledge practically and effectively to reach broad operational and strategic objectives. That requires a clear understanding of what is meant by knowledge. We must be specific about what knowledge is to manipulate and judge how it affects people, culture and other factors within the enterprise and its environment. We must distinguish between what we mean by “knowledge” and “information.” At first, it may appear as mentioned in the graphic above that there is a continuum from signals to data to information to knowledge—and onwards, perhaps to wisdom. However, when examining the nature of these conceptual constructs and the processes that create them, we find break that make information fundamentally different from knowledge. Most people think of knowledge as a recipe to deal with a concrete, routine situation. However, few situations are repeated. Because of that, knowledge must provide us with the capability, the understanding in the form of mental models, scripts, and schemata concepts, and mental models. The discontinuity between information and knowledge, referred to above, is caused by how new knowledge is created from received information. The process is complex. To become knowledge, new insights are internalized by establishing links with already existing knowledge, and these links can range from firmly characterized relationships to vague associations. Prior knowledge is used to make sense of received information, and once accepted for inclusion, internalizes the new insights by linking with prior knowledge. Hence, the new knowledge is as much a function of prior knowledge as it is of received inputs. A discontinuity is thus created between the inputs and the resulting new knowledge. The resulting knowledge and understanding is formed by combinations of mental objects and links between them and allow us to sense, reason, plan, judge, and act.

2.3 Definition of Knowledge Management

The following statement gives a definition about what could be understood as knowledge

“Yesterday’s data are today’s information, and tomorrow’s knowledge, which in turn recycles back through the value chain into information and then into data.” [Spiegler 2000]

This statement should demonstrate what we are talking about when using the term knowledge management system. The problem our days is that terms like information system, knowledge management, information retrieval or even knowledge database are often used in a questionable context as those are building keywords in the new information age.

The usage of those key- or buzzwords has both positive and negative effects. The positive is that people start creating images and ideas what buzzwords represent, which means that they get used to the topic and are not afraid of their use anymore. The negative effect is that those words are being introduced by marketing and sales departments. Those impressions often distort the real meaning. Hence people get an imagination of those systems which can be far away from reality. For instance many believe that a data warehouse is just another view to the real database. Some people do still not understand that the concept and the design of such a database is fairly different from that of a normal database.

Also knowledge is sometimes not distinguishable from data and information [Alavi 1999]: “At the beginning we just had data and information afterwards we had data management and information management. Technologies like databases, data warehouses and information retrieval systems were introduced. Now we have knowledge management and knowledge becomes a traded good”.

2.4 Aspects of Knowledge Management

2.4.1 Introduction

Lately knowledge becomes - besides other productivity factors like labour, capital and land - an important good [Abdecker 1999]. Nevertheless knowledge is a neglected good as it is difficult to discover and still not a hard good like the other factors. It is stored in individual brains or implicitly encoded and hidden in organizational processes, structures, documents, services and systems. Sometimes undiscovered knowledge is hidden in large databases or data warehouses. To discover such hidden coherences in databases data mining is used. Data mining and data warehousing can fill a book on its own and will not be explained in detail here.

To protect intellectual assets from decay and to improve the process of decision support knowledge management becomes a discipline of its own. To survive in the market today every enterprise should consider to have a suitable knowledge management system. For that KM has been recognized as one of the key factors of a future enterprise. Especially the organisational technological and social changes justify the use of such a system.

Computer scientist from different fields carefully pay attention to this new area and try to develop computer systems to manage the new knowledge factor. Many systems failed due to low participation, others had little success [Krallman 2000]. One noticed that there is probably no all-round solution and a software implementation always goes with some compromises.

The effective and efficient use of knowledge through a professional knowledge management system is the basis for a successful new cognition which could improve workflow processes.

At the present moment the young discipline knowledge management can be seen from three different points of view which are outlined in the next sections [Krallman 2000].

2.4.2 Informatics and Knowledge Management

The work done in the field of informatics focuses on the use of information and telecommunication applications, considering knowledge management aspects. The complexity of knowledge and its requirements to handle it perfectly exceed the possibilities in that sector of informatics at the moment. On the other hand a lot of new technologies have been developed which are succeeding in individual fields of knowledge management. Database systems, applications of artificial intelligence, the principle of the object orientation, data mining and other knowledge tools are some examples of useful implementations in the certain branch of knowledge management. These tools support beside the communication aspects the storage and the distribution of knowledge. However, these aspects build just a part of the knowledge problematic. As mentioned before the semantic aspect of knowledge the question about the value and the meaning of knowledge are not solved yet. For that reason it is necessary that the enterprises' intranets, knowledge databases and km-tools identify knowledge sources to extract, store and renew them.

2.4.3 Business Practice and Knowledge Management

More than ever, companies are confronted with a dynamic environment which changes permanently and fast in an extent and speed which was not known before. The continuing increased dynamic of competition makes it necessary to rethink actual success strategies. Different technologies grow together and the increased relevance of information and communication technologies cause shorter production cycles and let arise hybrid products. Even new competition sectors like E-commerce come into existence. The global spread out and flood of knowledge occurs very fast, but on the other side the half-life period of knowledge is becoming shorter and shorter. So called knowledge workers will play a more important role. Those persons are responsible for keeping the knowledge up to date and are responsible that knowledge becomes a key factor.

The idea described above, are especially important valid for service enterprises. Especially the management consultancies recognized the importance of their knowledge very early and those companies were the first which showed activities in that field and developed software for storing and retrieving knowledge [Krallman 2000]. But also production companies realize the importance of knowledge, and became sensitized to that subject.

In many places companies recognize that a efficient improvement of the production cycle is not a guarantee for a successful business, it can only be another piece to support the

companies success in a turbulent environment. It should be more important to have and use strategic flexibility, creativity and business energy, to conserve renewal processes. Renewal again is based on knowledge from actual facts-knowledge till strategic knowledge. Based on this thesis knowledge becomes a new value as a “soft good” which can be manifested in the company in many different forms. New job sectors arise like Chief Knowledge Officer who’s task is to maintain the knowledge and to find new resources of interest. The Chief Knowledge Officer could also declare the value of the knowledge which associated with the company. Especially the enterprise value of a consulting company is based on knowledge which can hardly be described in numbers or cash. More or less the only cash value that can be measured which reflects the value of knowledge are the costs being spent for software or education. By now knowledge will be accepted as a central resource in the enterprise which has to be managed from different perspectives.

2.4.4 Strategy Research and Knowledge Management

The actual researches being done in the field of knowledge management have a lot of positive resonance. In practice people are isolated engaged with the issue knowledge and its management while in the research views the resource knowledge is closely combined with dynamic learning. Knowledge from that view is always seen as a innovation and a resource with its own characteristics.

Mechanisms and approaches to learning have been explored throughout historic times. In recent years a formal learning theory has emerged to provide a framework to understand learning. Learning theory postulates that all learning models consist of four principal parts [Wiig 1999].

- a class of languages or other structured means of communication, one part of which is the “target of learning” (material to be transferred)
- the learning environment that provides knowledge material to the learner
- a learning strategy that maps new knowledge material onto hypotheses based on prior knowledge; and
- a success criterion that defines acceptability and correspondence between the learner’s conjectures and the learned material.

Unfortunately, learners differ considerably in both backgrounds and cognitive styles. This diversity requires that the teaching-learning processes must be versatile to take advantage of the learners’ strengths and compensate for their weaknesses.

2.5 Challenge of Knowledge Management Systems

2.5.1 Challenge of Knowledge

Companies do no longer compete as much by resources and the economic use of them as they did ten years ago, as this market has reached a stable form. More and more we can see that companies differ from each user by the deployment of the resource knowledge. To underline the importance of the resource knowledge in our days this term will be described from different points of views. Lately many efforts from researchers, consultants and managers were made in the field of total quality management, lean

management, business process management, reengineering or the implementation of a learning company. By now companies discover knowledge as a challenge, both in an academic and economic way. This is the step towards a knowledge society. Not the production of material goods is the central challenge of an enterprise, but the solution of problems in a fast changing market. When companies reduce their physical or material production their creative and knowledge intensive activity grows. The main part of tasks in new companies is now in the field of knowledge work. This work could be described as tasks that are based on specialized expertises of persons, who have to adopt their profession in lingering training processes [Willke 1997].

For enterprises now exists the problem, that on the one hand they need a relatively high amount of intern specialists and on the other hand the knowledge of these experts has to be associated to a demand or service in the market. The result of that development is, that the deployment of work and capital resources are being shifted in favour of a better use of the resource knowledge.

In the industrial time the main focus of the companies was not to waste working capacity. In the years of automation they tried not to waste the resource of machines. By now they try to improve the use of the existing information.

2.5.2 Knowledge as Resource

Besides the traditional resources like raw materials, work and capital, we talk about another resource, as one could guess it is knowledge. This can be explained in the following statement. "Knowledge can similar like other production factors be analysed, balanced and managed." [Probst 1997]. Such a point of view has crucial consequences to the treatment of knowledge. It should be one's aim to dispose this resource economically and usefully. This could for instance mean to get it on time and to use it without restriction at the location where it is needed. This means the right knowledge at the right time at the right place.

To reach that statement, one has to make an effort for the creation, codification and the storage of the knowledge to manage and control the new resource.

Knowledge is not a resource in a conventional sense like for instance raw materials. There is no way to consume or exhaust this resource. Shortness of knowledge does not exist, only if the source to the information is hardly accessible or if the information needed is costly. Through division new knowledge can arise. But also knowledge can not be seen as a good that can be warehoused, divided and transported arbitrarily [Probst 1997].

2.5.3 Changing Established Habits

Over time knowledge takes the risk to become a blockade for further or future learning. New things are always associated with a new step into suspense, uncertainty and frustration. The willingness of learning new knowledge depends on the degree and acceptance of disappointment of the individual. One knows something until she will be convinced of the opposite. Nevertheless the temporal and sometimes even approximating validity of the knowledge will be likely overseen. The more often a knowledge or a rule has been approved the harder it is to let it go and to accept innovations in this area. We all know when somebody says that they always did it that way and why they should change their established habits, if the old process was satisfying and lead to a result. People are afraid of innovations as they are not used to it and because of the fact that

something new is also a degree of uncertainty. This has the effect that new thoughts and innovations are not tried out. On the other hand does this “branded knowledge”, if it was often enough employed successfully, build a standard in the companies knowledge database. This normalisation and standardisation ensures that processes do have a stable sequence, so that the factors insecurity and disquietude are being eliminated.

ISO 9000 for instance is a standard to improve the quality management of a process. You choose to follow this path of the guidelines because you feel the need to control or improve the quality of your products and services, to reduce the costs associated with poor quality, or to become more competitive. Or, you choose this path simply because your customers expect you to do so or because a governmental body has made it mandatory.

The consequence is that know how that is branded in peoples mind could hardly be changed as the willingness is not given and the effort cannot be seen. Employees only see the additional expenditure which is necessary to learn something different. The benefit of it might be too far away. This is an important factor for almost every new technology as well. Managers are convinced of a new process as they can see the benefit of it, but they forget that their workers often do not have the same information as they have. The conclusion is that before deploying a new system the installers have to promote their product to the people who really work with it. Otherwise they take the risk that the system will be refused which has the opposite effect it should have. The worst case would be that a innovation or knowledge expansion causes the enterprise to be instable.

The preparedness to learn and the quitclaim of stable knowledge is associated with insecurity and the result of the learned material is because of that uncertain, too.

From that point of view knowledge and learning are complementary terms. Either something is known or something has to be learned or as we can also say: either expectations are sustained or they will be rebuild. The circle of that view can be closed when we consider that learning is the requirement of knowledge while knowledge is the limitation of learning. The willingness of learning is a permanent process of surprises. This can be a learning processes where the result of the process is almost unknown. Even sometimes the result can be, not to learn anymore.

2.5.4 Communication of Knowledge

Knowledge that stood the test in one department is not necessarily relevant or applicable in another department. Mostly everything depends on the individual system history and the matching of the knowledge to experiences. Communication supposes that the receiver can handle the content and that the information is relevant for him. Especially in that case a question answering system like the one developed for this paper has a big advantage. Only knowledge which is desired and relevant is communicated. We imply of course that the knowledge or rather the answer of the question is correct und understandable.

Knowledge that will be transported as knowledge, always includes the fact that the receiver does not know anything or maybe not enough of the actual subject. Or in other words which system likes to reconstruct itself only because another system or individual claims that it knows something new or better without knowing if the effort to learn something new is really successful and brings you to a higher step.

From that we can see that the open and the covered rejection of knowledge can also be a result of the knowledge effort as well as its adoption. A good example for that is a rule

which is being added to the Cyc (see chapter CYC) database which is evidently false. For instance that cars drive 400 km/h. The rule is a kind of rejected knowledge.

That can be a drawback of the transfer of knowledge which is rarely seen by the classic approaches of knowledge transfers. [Götz 2000]

Comprehension and Cross-Border Communication

As already explained in the section above, knowledge depends on the experience of an individual and the system history. Experiences, information and answers that were successfully approved are stored as positive and impressions that caused failures were stored as negative. Every information that will be added into a system has to be processed so that it has a certain context. The easiest and most efficient way to transfer knowledge is to communicate it.

Especially organized companies are designed to work autonomously and to communicate from time to time while holding a meeting for instance. Besides that communication in between that structure is very recommendable. Moreover the concentration to one's work and environment has the effect that work can be done very efficiently.

Knowledge is being build in between a certain frame of reference and only makes sense in that surrounding. A frame of reference will mostly be a department or a project group. Most of the communication takes place in that closed group and will only be transmitted cross the border in a well defined way which has been discussed in advance. The frames of reference do not fit together as one department is not informed of the operation mode of another department. This is a typical example where the resource knowledge is wasted or not used efficiently because of disinformation or a lack of communication.

The implementation of a question answering system could close the gap and build a bridge between hierarchical splitting as questions are visible for everybody independently of his position and location. New networks of specialists can be found which have been separated before because of weak communication. One secret of an successful company which survived in the market is exactly that the drawbacks of the hierarchical structure are compensated through a department overlapping knowledge interchange. A question answering system like ours can fit perfectly in that policy.

A problem about sharing knowledge can be that the sender of information is afraid of losing his monopoly and with it the security of his own job. If the process of offering knowledge will be rewarded the knowledge owners loose the fear of becoming unimportant. In addition if they recognize that the disposal of their own knowledge creates a discussion which on the other hand could improve their own view of things, they will pretty soon participate and like that concept. How an individual absorbs and stores new learned knowledge depends on the personal attitude and can hardly be manipulated. Another advantage can be to receive respect from your boss and other employees when they recognize that you are a knowledge owner and willing to share. It is important that the department manager takes away the fear to share knowledge. The predication has to be clear, we are one team and nobody is working for his own.

2.5.5 The Management of Knowledge

As the knowledge oriented society is by its means a society of organisation, the main focus has to be the management. The term management has to be explained by here. When we started to adopt the term management the aim was to explain management from the organisational point of view. The reason was that big companies built their first organisations. After a while it became clear that the management is the qualifying organ

of a company. All companies need a management no matter how they call it. Often you hear a manager saying that it does not matter for which company she works. One could work as an IT or an automobile company. The tasks and the problems are almost the same. Management is a abstract task and kind of independent of the product and the services the company offers. Because of that we can say, that all managers do more or less the same.

The main task for every manager is to bring together people with a certain knowledge with other people to join their services which can improve the companies benefit and to create new services. That means that managers need both, the knowledge of their daily work and the understanding of the companies structure. The main focus has to be on the companies' markets, its abilities, its values, the environment and of course the essential capabilities. [Drucker 1996]

The theory of management is a common science of the formation of social systems. The feature of a social system is, that it is composed of many subsystems which again can be humans or other subsystems. The science teaches us how to perceive these systems cognitive and how to divide them in our thoughts in smaller components. Another point in the science of management is arrangement and formation of organisations by means of a destined social system. Usually those systems include many agents which are allocated to a special function or role. Through various mechanisms of coordination it will be tried to reach the aims which have been chosen before. Two different forms of coordination are distinguished. Firstly the hierarchy and secondly the market. The agents of a market are autonomous and decide freely if they take part in a equity switching. The hierarchy is based on the consolidation of resources which causes a loss of autonomy. Those vertical hierarchies are called organisations which can be firms, associations or clubs.

A second field in the management theory deals with the figuration of processes. It is tried to have a strategy to reach a certain aim. Those aims will be subdivided in several concrete flows and will be supported with the necessary resources.

Management could be seen as a free decision to take the responsibility for a clear defined aim. To reach this aim resources like humans and raw materials or machines are used in a work process where the power of influence of the management is used to control these processes. [Hoffmann 1989].

In the real world this means that an individual has a contract with a company which fixes his income for benefits made for the company. The top management will be responsible for any failures, which means that if the management is not able to manage the processes named above it will be fired in the worst case. As nobody can have an overview of all processes in a company it is important that the information channels are working properly. Hence it is the beginning of this subchapter. It is important that managers get an overview of their departments to bring together people with a certain knowledge to improve the companies benefit. Many systems are used to give managers a compressed overview of the companies status quo. MIS (Management Information System), Statistics, OLAP tools are just a view. Hopefully in a couple of years a question answering knowledge base is also an important management information system.

Competent managers distinguish themselves not only by their knowledge. More important are their social skills. The ability to lead and handle human resources is much more important than the perfect deployment of raw materials for instance.

2.5.6 Knowledge Management

The term knowledge management has a lot of definitions, which shows how manifold the domain knowledge management is. The formulations which will be explained here is mainly taken from the German literature.

It is the goal of knowledge management to bring the potential of knowledge which exists in a company into line, so that it builds a integrated and company-wide knowledge system, which insures a efficient knowledge processing to reach the companies aims. Based on that, the formation of the companies' whole knowledge and the deployment of natural and artificial resources for the knowledge administration is necessary. [Albrecht 1993]

In that chapter we have seen knowledge from many different points of views. We can say that knowledge is one abstraction level above information and we found out, that discipline knowledge can be subdivided in three different parts, based on informatics, business practice and strategy research. Later on we tried to see knowledge as a resource and found out that it is hard to change established habits of employees to teach them new processes. At least we were discussing the communication of knowledge and the problems that can occur.

3

Information Retrieval and Processing

3.1 Stemming

3.1.1 Introduction

Stemming indicates the building of stems from a word. If a document management system uses stemming it has not only the ability to look up the word during a full text search, but also the conjugated form of that specific word. For instance if someone searches for the word *drive* the retrieval program should also discover locations with the word *drove*.

Although everything in our world and in our times is Multi- and Hypermedia, text is still the preferred type of encoding and saving information. More than 80% of the documents found in Google or Altavista are textual [Bray]. Search engines like Altavista claim to have indexed about 100 million documents. These two facts imply that everyone who has access to the internet is overwhelmed by the number of documents being offered. Without a reasonable indexing scheme it seems to be impossible to get a survey. As no one is able to keep track of that amount of data and nobody could index these files manually, there is a strong demand for automatically indexing, searching, extracting and integrating information .

3.1.2 Stemming and Decompounding

One of the major problems for any kind of automated text processing is the detection of different morphological variations of the same word. Especially for question answering systems, information retrieval and data mining, these variations must be detected and mapped to a common form. Most algorithms try to find the word stems to compare and save the words in a common form. The problem about stemming is, that a stemmer-program should conflate together all and only those pairs of words which are semantically equivalent and share the same stem. This aim is hardly to reach perfectly as a stemmer-program is at our days not able to work faultlessly. But actually the error rate can be decreased to a negligible minimum depending on the algorithm used.

Stemming is the building of the stem of a word. If a document management system is able to manage stemming, it can not only find the typed word, while using a full-text search, but also its conjugated or declination form. For example if someone searches for *to find* the according retrieval program also finds the reference source with *found* or *finding*.

As described above many question answering systems (QAS) or information retrieval (IR) systems which dispose some kind of stemming. If a system uses stemming the aim

is to join derived words together to a common stem. The side effect thereby is the reduction of the size of the search index, but more important is the allowing, to retrieve documents independently of the specific word form used in the query and documents. The main reason for the use of stemming is the hope that through the increased number of matches between search terms and documents, the quality of search results is improved.

Precision and recall are the two most important measures for a good retrieval system. Stemmed terms retrieve additional relevant documents that would have otherwise gone undetected. This leads to an improved recall. There is also potential for improved precision, since additional term matches can contribute to a better weighting for a query and document pair. If the stemmer-algorithm joins terms too aggressively, many extraneous matches between the query and irrelevant documents are produced; this is called "overstemming". Even though the stems that are produced may be correct from a linguistic viewpoint, they may not be helpful for retrieval. In contrast, if important relevant documents are missed because of a conservative stemming strategy, we speak of "understemming". A good stemming-algorithm has to find the right balance of combination for effective retrieval.

Sometimes we distinguish weak and strong stemming. By using weak stemming only different declensions of words are detected and conflated. For instance the word "brothers" is stemmed to "brother" and the irregular word "mice" is stemmed to "mouse".

If you use a strong stemming all suffixes and depending on the algorithm sometimes all prefixes are removed from the word. For example the word "illness" is conflated to the adjective stem "ill".

3.1.3 Stemming German compared to English

Thinking about our question answering system we concentrate on weak stemming as it is sufficient, as in many cases prefixes and suffixes carry lexical and semantic meanings. There are some problems that can occur by using weak stemming, which of course causes errors. The aim is to reduce these to a minimum. There is the problem to detect the boundary between stem and ending, by cutting of too much of the stem it loses its meaning, by cutting off too little it won't be the stem of the word. The most difficult part is to find the regular changes of a stem. Like in German there are words like "Haus" (house) and "Häuser" (houses). In comparison to regular changes to the stem there are also irregular changes like "to go", "went", "went" or in German "nehmen" (to take), "nahm" (took) and "genommen" (has taken). This problem as well can only be solved by implementing a lexical functionality. This problem can only be solved perfectly, by supporting the algorithm with a lexicon-function. This, on the other hand is very time intensive, as every word has to be looked up. Another problem is the so called glue character. This occurs by words like "swim" and "swimming" or in German for words like "essen" (to eat) and "gegessen" (have eaten).

Languages like English with just a few morphological variants, the most common stemming method is suffix stripping. The idea of suffix stripping is to iteratively cut off all suffixes from a word. Suffix stripping is usually based on a dictionary of suffixes, a longest match algorithm, and some simple morphological rules. An algorithm in common use, which uses that kind of suffix stripping was invented by Porter [Porter 1997]. Most search engines in the web use the strategy of Porter to retrieve their documents [Porter

1997]. The implementation of the prototype QAS of this thesis will use the Porter Stemmer Algorithm to index English texts and documents, too.

3.2 Stemming German

3.2.1 Compounds

By now science brought up different methods to retrieve documents or texts ranging from language-independent to sophisticated linguistic analysis. This chapter takes a closer look at the German language, where, besides stemming, decomposing seems to be an additional issue in retrieval.

Compounding is the combination of two separate words. The English language has little of those words in comparison to German. It seems that most studies about stemming are dealing with the English language. This rules can't simply be adopted into German, as the German language uses this compounding words quite often. For that reason if someone tries to stem in German a more sophisticated approach is needed as the declensional structure is more complex than in other European languages. Nevertheless I would like to exemplify the principles of a stemming algorithm in German as the prototype will offer the indexing of documents in German, too. The prototype will hereby use the algorithm described below.

In most Germanic languages, but also in some other languages like Finish, it is possible to build compounds by joining several words together. By using such words called *compounds* the performance and especially the recall, may be negatively affected if the system does not take this word formation process into account. If the system is not able to decompose or analyse such compounds lexically, relevant documents can rarely be found. There are different approaches to analyze and split compounds. The process for decomposing the joint words is called *decompounding*. An example for an word from that domain would be *speedboat* (*german: Schnellboot*). The decomposed form by all means would be *speed*(*german: schnell*) and *boat*(*german: Boot*). We focus of our attention on the usefulness of compound splitting for information retrieval, as that is the work that has to be done in a question answering system. One has to be aware, that the splitting of some compound words can cause a shift of meaning. This is often the case for complex technical terminology. Although it is almost impossible for a computer to make a difference between words that should not be separated and those which should, one has to care about that factor. It is better to use a so called conservative decomposing algorithm to leave such compounds intact than trying to split a word and risking to loose its sense. For instance, if someone separate the word "backbone" almost everyone who is acquainted with the new technology knows that someone talks about a high performance wire that connects various networks. By separating this word to "back" and "bone" the meaning of the word and its interpretation gets lost and so the precision and recall.

The problem about German compound formation, which allows us the formation of new compounds straight forward, makes it impossible to correctly split all potential compound words. Another solution to that would be "aggressive decomposing". That algorithm produces a maximum number of decomps by using heuristics. I already named the problem that occurs with that mechanism. Especially as our project is being designed for a technological environment decomposing should be used- if at all- in a

conservative way. Most studies on stemming and decomposing behaviour have been conducted using small test collections [Moulinier 2000]. The size of the collection, especially the length of the retrieval items, is important, since short documents, for instance only titles and abstracts, increase the likelihood for word mismatch if no stemming is used. Therefore it is not immediately obvious that a performance improvement measured on small collections with short documents can lead to an improvement on larger collections with longer documents.

3.2.2 Characteristics of the German Language

German, in contrast to English, is a highly declensional language, a fact expressed by a rich system of inflections and cases. Depending on the word class, i.e. noun, verb, or adjective, there is a set of possible inflections for each particular word. There exist up to 144 forms for one verb, which makes stemming more complicated compared to languages such as English. Furthermore, words can be formed by attaching multiple derivational inflections to a stem in order to build new forms. For instance, the lexeme "inform" is the stem for "informieren", "informiert", "informierte", "informierend", "Information", "Informant", "informativ", etc. Additionally, in most Germanic languages like Dutch and Swedish and also in some other languages like Finnish, it is possible to build compounds by concatenating several words, such as for example, "Haustür" (house door) or "Frühstück" (breakfast). In almost all languages such compound formation occurs, such as "hairstylist". However, in English and in Romance languages these words are lexicalised, i.e. they cannot be expressed by a nominal phrase the way compounds in Germanic languages could be ("Haustür" vs. "Tür des Hauses"). Because Germanic languages also know lexicalised compounds, the treatment of such words is quite complicated. The success of compound analysis depends more on linguistic knowledge than stemming does. In principle, only words with certain types of part-of-speech can be coupled, for instance noun/noun, adjective/noun, or verb/noun. Some analyzers split only those compounds where the constituents have the same part-of-speech (noun/noun, adjective/adjective), and could be thus classified as "conservative". Others split the compounds into all possible word forms, often by means of a lexicon lookup. Because these methods do not consider the part of speech of the constituents, for instance they also split pronouns, prepositions or articles, this approach can be described as "aggressive". Linguistic compound analyzers lie in between, they are splitting compounds only into valid word forms, for example - nouns, verbs and adjectives.

3.2.3 Stemming and Decomposing Approaches

The following section of text represents different methods of text recognition ranging from completely language independent methods to components that use elaborate linguistic knowledge. Which procedure to use in practice, depends on the system to be implemented and the individual preferences. The presented algorithms should just demonstrate the different trains of thought.

Combination of word-based and n-gram based retrieval

The use of combined character n-gram and word-based indexing was reported as a successful approach to German text retrieval by Mayfield [Mayfield 1999] and Savoy [Savoy 2002]. The individual 6-grams, built on the unstemmed words, potentially span

word boundaries. A main benefit of this approach is its complete language independence - no specific linguistic knowledge is needed to form the n -grams, and the word-based indexing is done without attempting conflation. On the other hand, the method is storage-intensive: the large number of different n -grams leads to a massively increased index size, which is roughly three times that of an unstemmed word-based index.

Linguistica: Automatic machine learning

Linguistica [Goldsmith 2000] performs a morphological segmentation based on unsupervised learning. The aim is to find the correct morphological splits for individual words, in a language-independent way. Possible categories of stems are identified using a set of suffixes that is detected solely based on surface forms. As an outcome Linguistica produces a lexicon comprising each word of the collection together with its possible affixes, for instance "machbar" (feasible), "machbar|en" (feasible), "machbar|es" (feasible), "machbar|keit" (feasibility). Decomposing occurs only by accident; i.e. if a word is frequently used as a compound constituent, the systems may incorrectly classify this word as an affix. Unfortunately, this means some compounds are conflated with only their modifier, i.e. "Datenbank" (data base) is conflated to "daten" (data), losing the constituent "bank".

NIST stemmer: Rule-based approach

The NIST German rule-based stemmer [Porter 1997] has been constructed by analysis of the frequency of German suffixes in large wordlists. The stemmer is available as a part of the NIST ZPrise 2 retrieval system. Its approach is similar to the Porter English stemmer [Porter 1997]. The rules were hand-crafted to produce as many valid conflations of high-frequency word forms as possible, while keeping the rate of incorrect conflations low. The stemmer attempts to iteratively strip suffixes from a word. For instance the word "glück|lich|er|weis|e" (luckily) is reduced to "gluck" (same stem as for "luck"). The stemmer can be combined with a corpus-based decomposing component based on co-occurrence analysis. After collecting a list of candidate nouns, the component tries to find valid splittings by looking for potential constituents that co-occur in the same documents. This purely corpus-based approach produces a number of errors, but is overall rather conservative in the number of splittings generated.

Spider stemmer: Commercially motivated (rule-based and lexicon) approach

This should be a short overview of the stemming component used in the commercial Eurospider retrieval system [Wechsler 1997]. One of the aims is that the whole project is Open Source, hence this part is just for inspiration. The approach is based on a combination of a lexicon and a set of rules which are used for suffix stripping and unknown words [Wechsler 1997]. This stemmer has been used for over five years in all commercial installations of the Eurospider system, and has therefore been constantly adapted according to customer feedback. The component includes optional decomposing, which can be applied in one of three modes, from conservative to aggressive splitting.

MPRO: Morpho-syntactic analysis MPRO

This is a development by the IAI [Maas 1996], performs a morpho-syntactic analysis consisting of lemmatization, part-of-speech tagging, and, for German, a compound analysis. MPRO uses general morphological rules in form of small subroutines which co-operate with a morphological dictionary. As result for each word a set of attribute-value pairs describing inflectional attributes, for instance. gender, number, tense, mood, word structure and semantics like a lexical base form, derivational root form, compound constituents, semantic class, etc. is produced. With this tool, the corpus has been analyzed and for each word, information about the lexical base form, and the derivational root is used to generate lexical resources. For instance, the analysis of the word "Kollision" (collision) results in {string=kollision,c=noun,lu=kollision,nb=sg,g=f,t=kollision, ts=kollision, ds=kollidieren~ation,ls=kollidieren,s=ation,...} and produces as a lexical unit "kollision" and as a root form "kollidieren" (collide); for the compound "Schiffskollision" (ship collision) MPRO generates a splitting based on lexical base forms, "schiff_kollision", and one based on derivational root forms of the compound constituents, "schiff#kollidieren".

3.2.4 Stemming German Texts

Stemming for morphological more complex languages like Dutch and German is not that easy. The reason why the combination of longest suffix matches and simple linguistics doesn't work properly have different reasons. One is for instance, that changes of the stem usually don't occur at the end of the word, but more often in the middle. An example would be again "Haus"(house) and "Häuser" (houses). Suffix stripping in that case never leads to the same stem. The next problem is, that all rules for declensions of nouns are based on gender. Without lexical analysis or dictionary search it is impossible to say whether "er" is a suffix like in "Bilder (neutral gender)" (pictures) or just part of the stem as in "Leber (fem.)" (liver). There are many exceptions in German for when to exchange a vowel for an umlaut to build its plural form. "Hunde" (dogs), for instance is the plural of "Hund" (dog), whereby Mänder (mouths) is the plural of "Mund" (mouth). The last problem about stemming German texts is the heavy use of compound nouns, as described in the section above.

At the present moment there is no fast algorithm available which combines both power and speed for stemming German texts. Linguistic approaches have difficulties with compound words which must be decompounded before stemming. Dictionary based approaches are slow because of there large word lists. Piotrowsky [Piotrowsky] argues that at least 50,000 words are needed to achieve meaningful results. That is the reason why big German web-search engines do no stemming at all. Because of the complexity and slowness the prototype QAS disclaims a lexical analysis.

3.2.5 Discriminating, Substitution and Stripping of German Text

The algorithm explained now is used in the Jakarta Lucene Projekt (for German), which is Open Source and will be implemented as the preferable stemmer. The algorithm is written in Java and is based on a two step technique. [lucene]

The prototype of the QAS being developed will make use of that software package which is Open Source and for that free of charge. The algorithm is based on a context free suffix stripping. Therefore the core algorithm is very fast and simple. It can handle an infinite number of words and gets even stronger the more compounding occurs. The only requirement for the algorithm is that all stop words as well as determiners and other function words have to be removed from the text before it can be stemmed.

The described algorithm is designed for stemming German words but it can easily be adapted for other languages. The used algorithm is not better than other approaches in use, but it has the advantage of being much smaller and faster. Another benefit is that it can easily be extended by either linguistic rules or wordlists to improve quality which on the other hand costs speed.

As already mentioned the purpose of stemming is to reduce different morphological forms of words to one common form. A discriminator is the stem of the word build by the analysis of the algorithm. The discriminator does not necessarily have to be the real stem of the word as long as it is unique and can not be build from other words. But all declensions of a word have to be stemmed to the same discriminator. In a real system it is almost impossible to reach those requirements of uniqueness. But the error rate can be so low, that it does not affect the system in a negative way. Usually word stems are unique if you do not think about homonyms. Especially the homonyms make it impossible to perform a perfect stemming. Without knowing the context of the sentence the mechanism is kind of lost, because a stemming algorithm can not differ between words having the same writing but different meaning. A basic approach of understanding the context of sentences has been described in the project CycOrg above [cyc].

Another problem might be the use of irregular verbs or declensions. Especially German has variety of verbs or nouns which do not follow a declension rule. To stem these kind of words a very sophisticated algorithm has to be used which discovers the irregularities. Until know, it is hard to find an algorithm which meets this requirements without a certain error rate. The alternative way is to use a dictionary to compare the irregularities and to find the stem of the word. This alternative is very time intensive and should only be used if the avoidance of errors is an important factor of the system, considering that this leads to higher needs of system and time resources. Without using a dictionary to compare the words, every stemming algorithm produces some errors. It is the aim to keep the error rate as low as possible. Even the best algorithm cannot avoid that some words do not follow the typical declension rules and in fact produces errors.

In the prototype being produces, it is relatively unimportant if the error rate is close to zero. It is essential that the results will be returned fast. The question is, if the error rate is low enough to accept the result. In most cases this is the fact.

In the first step, characters- or certain groups of characters- will be replaced in each word which will be stemmed. The second step cuts of the suffixes regardless of the context of the sentence. The problem could be, that different words map to the same stem. In our

case this problem causes only more results to a question as the same stem leads to a higher match of answers. Moreover the error rate is relatively low, especially considering that the use of the question answering system is usually used in a technical domain. When considering technical text, those terms are often abstract, so that they rarely cause the same stem.

3.2.6 Character Substitution

While stemming is mainly based on prefix and suffix stripping which will be described in the next section, another important procedure is the substitution of characters. Straight German builds many plural forms by changing vocals in the middle of the word. For example “Haus” (house) will change to “Häuser” (houses) in plural. This effect can only be handled by changing the umlaut to a common form. For that reason every Umlaut will be transformed in it’s associated vowel. The same happens with the character “ß”. It will be transformed to an “ss”. Another problem is the occurrence of irregular verbs like “halten” (hold) and “hielt”(held). The irregularity can only be managed by changing certain character combinations into a common form, so that the irregularity will be circumvented. Characters that belong together because they build a single sound like “sch”, “ie”, “ei” are split apart and will be cut off by the algorithm. These characters will be substituted by special characters. “sch” changes into “\$”, “ch” will be transformed into “\$”, “ei” into “%” and “ie” into “&”. If double characters occur like “ss” or “tt”, the second character will be changed into a “*”. All these character substitutions are done to save them from pre- or suffix stripping. The main advantage of this substitution process is, that most plural forms will be stemmed correctly and correspond with their stemmed form in singular. The main drawback is that some words usually not belonging together now build a common stem. To name one example the word “Eisbär” (polar bear) will be substituted to “Eisbar” (ice bar), which definitely is not the same. But these errors are so little that one can easily forget about it in most applications.

If wanted some other substitutions can be implemented with little effort. By recognising that a certain word or more words will permanently build the wrong stem, a character substitution might help to avoid this stripping-error.

The only requirement for any substitution is, that the new word should either result in another morphological form of the same word or produce a completely new word. If that fact is not met sooner or later a stemming error occurs.

Substitution can be a very useful procedure as it can capture linguistic rules and statistical heuristics. If for instance, “ge” within the character sequence “ige” can never be an infix denoting participle, “ig” should be substituted from being split. Most of the substitutions are short and can be implemented easily and efficiently.

3.2.7 Suffix-Stripping

A technique which is used in almost every stemming algorithm is suffix-stripping. By using suffix-stripping each word is matched against a list of suffixes. One starts with the longest suffixes first, so the longest suffix matching the word’s end will be cut off. For instance the word “walking” would match the suffix “ing” which would be cut off, so that walking will be transformed to “walk”. As we do not care about the context of any sentences this method is called context free suffix stripping. Of course this method of course is not very new and is claimed to be very error prone. With some additions(which will be explained later on), it will be tried to reduce this errors to a minimum. The point

is that removing the suffixes sometimes or even more often does not map the original stem of a word. As long as the stemming algorithm is only stemming this particular word and its declensions to the specific form the word is uniquely comparable and fulfils the requirements completely. For instance “equal” will be stemmed to “eq” which definitely is not the original stem of “equal”. Anyway there should be no other word which leads to the form of “eq”, so that “eq” is useful for comparing it with other words.

To remove the suffixes we have to find all the suffixes available in a language. The German language has the following suffixes

- 7 for nouns (s, es, e, en, n, er, ern)
- 16 for adjectives (e, er, en, em, ere, erer, eren, erem, ste, stersten, stem)
- 48 for verbs (e, est, st, et, t, em, ete, te, etest, test, eten, ten etet, tet, end, nd...).

By „adding“ *end* or *nd* at the end of a verb it will be turned into an adverb.

After a verb was turned into an adverb any of the suffixes of an adjective can follow so that 48 possibilities can be created. To speed up the process it might be possible to forget about some suffixes which will be used little, especially those which include each other. The gain of it might be big, compared to the new error rate. Finally any occurrence of “ge” both as prefix or suffix will be cut of.

To make the algorithm more sufficient some context base consideration will be made. Especially the length should be taken in consideration as it makes no sense to cut of a letter from a word existing only of two letters. For that reason words not longer than four characters will be left alone and further stripping will not occur. The second restriction is that if a term is shorter than five characters neither “em” nor “er” will be removed. The third restriction is that if a term is shorter than six characters no “nd” will be removed. The fourth restriction says, that “t” will not be removed from terms starting with an uppercase letter. The last rule is simple to explain, as “t” is an verb suffix the removal of that character is of no use for nouns. One has to know of course that all nouns in German start with an uppercase letter. The negative side effect of the four rules is that words with two letters will be ignored by the algorithm. As there are only a few of this words in German this effect does not play an important role at all.

3.2.8 Evaluation

Two problems that occur with every stemming algorithm. They have been mentioned before and are the problems of over- and understemming. If overstemming took place two different words which normally do not belong together where stemmed to one common stem. On the other hand understemming happens when two words originally derived from a common stem were not stemmed two a common stem. It is the aim of the algorithm to find the right stemming method to minimize or balance under- and overstemming.

3.2.9 Understemming

Most of the time, the introduced algorithm strips of too many characters of a certain word. This is not a disadvantage as the singularity of the stem will be preserved. There is also a special group of words where the algorithm tends to the error of not removing the whole declension suffix. One appearance of that words are irregular verbs and all of their compoundings. Most of these irregularities lead to different stems. For instance in

German there are about 173 irregular words which are being used frequently. This error can only be avoided by using a dictionary which of course might be very time intensive.

Another representative sample is the female form of a profession. Most of the time this forms do not lead to the same discriminator. An example would be the word "Schauspielerin" (actress). This word will not be stemmed at all while "Schauspielerinnen" will be derived to "Schauspielerin".

Another category of words causing an stemming error are words with a Greek or Latin origin. Two examples are "Drama" (drama) where the plural is "Dramen" (dramas) or "Minimum" (minimum) where the plural is "Minima". Both words do not have a common stem after the algorithm was used on them. This effect again can also be avoided by looking up in a dictionary for the right form.

The same counts for many inhabitants of countries, for instance "Spanierin" and "Spanierinnen". The same with inhabitants of cities. To counteract against the error caused by female forms of words, one can substitute the ending "erinn" with that of "erin". Especially because in our days the female form of a profession is more polite than naming the male form, one should decide if the substitution causes a better stemming result, considering that the little effort uses certain amount of computing time.

It is mentioned that cases where two different morphological forms of the same adjective or noun are stemmed to different discriminators are very seldom. Most of the verbs we use are regular and for that easy to stem. The problem about the irregular verbs is, that they are very common in the German language, both spoken and written. While taking into account the percentage of verbs being irregular we notice that only five percent of all the German verbs are irregular, but while looking at a common text sometimes those words are used up to 50 percent.

3.2.10 Overstemming Nouns

Overstemming occurs when different words are being derived to the same stem although they have a different meanings. Hereby some examples should be listed which regularly could cause overstemming errors. Known and difficult problems are names like "Albrecht Dürer". This name will be stemmed to "Dur". This mistake is very hard to avoid also with more sophisticated approaches, as not every proper name is listed in a dictionary and names do not follow any declension or suffix rules.

Other difficulties occur by words with overlapping stems like "Rind"(beef) and "Rinde"(rind) or "Eis"(ice) and "Eisen"(iron). Especially stems ending on "er" tend to cause overstemming errors. This words with a overlapping stem can only be handled correctly if the gender is known. As in our case no context detection will be used a gender recognition is impossible especially because gender stripping is extremely difficult in the German language.

Derived forms from the same stem are a potential error prone. "Tanz" (dance) and "Tänzer"(dancer) are words of that category.

Umlaut substitution frequently can lead to a misinterpretation. The word "Stück"(piece) is turned into "Stuck"(stucco) as well as "Bär"(bear) is turned into "Bar"(bar).

The last noticed overstemming error can be watched by words like "Wien"(Vienna) or "Wiese" (meadow) as both have the same discriminator. The same with "Hafen"(harbour) and "Hafer"(oat).

Most errors occur by words being shorter than six characters. It can be proved that the longer the word the better the stemming success. In most cases the number of overstemming errors is lower than the number of understemming errors. The reason for this behaviour normally is the high frequency of irregular verbs in written texts.

3.2.11 Improvements

Irregular verbs which lead to over- or understemming errors can easily be handled by adding a irregular verb wordlist at the substitution part of the algorithm. This of course causes extra computer resources and longer waiting periods, but by extending the algorithm with the most frequently used irregular words, the error improvement should compensate this drawback. The idea is to reduce all morphological forms of an irregular verb to one of these forms by using a small wordlist. The common form should be the most significant character sequence of all forms. For instance the discriminator form of “kommen”(to come), “kam”(came) and “gekommen”(was coming) is “komm” whereas the discriminator form of “laufen” (to run), “lief”(ran) and “gelaufen”(was running) is “lief”. The word does not necessarily be similar to the real stem it is more important that the discriminator is unique.

The improvement is done by simple substitution of the associated word. Even occurrences in a compound word are treated like a irregular form. For instance the word “ankam” (to arrive) will be changed to the discriminator “ankomm”. This makes sense as irregularities do not only occur the normal word but also in all compounded words where the irregular verb is included. By using a small wordlist the understemming error prone can be reduced significantly.

Tests with the algorithm showed, that the stemming error rate is normally less than five percent. This should be acceptable for our prototype as well as for many applications [Rijsbergen].

A very interesting part is also the very low error rate, compared to high sophisticated algorithms. When concerning that the algorithm is very compact and fast this behaviour seems all the better. The algorithm represented here can be implemented in its easiest form with round about hundred lines of source code. Compared to algorithms which refer to dictionaries of more or less 50,000 words the procedure introduced here his incredibly workable. Furthermore the algorithm could be used as a extension to an existing wordlist based approach.

Other algorithms using wordlists have the mayor drawback of being slow, but the advantage to have an error rate close to zero. When trying to combine the described algorithm with a wordlist algorithm, while reducing the wordlist to one fifth of the most used words an very low error rate with a much better pass-through time could be reached.

The major drawback of the described algorithm is the high error rate for verbs and adjectives. But on the other hand it is a good, compact and very fast alternative to existing high cost algorithms. This approach is optimally applicable for the interactive systems where user enter questions in a natural language. In addition the algorithm is free of charge as it is part of the Jakarta Open Source Project Lucene [lucene].

3.3 Stemming The English Language

Stemming the English language is much easier than almost any other languages. The Porter Stemming Algorithm [Porter 1997] is an easy and fast way to stem words or terms written in English to a common form by suffix stripping. This algorithm will also be supported in the Lucene project [Lucene]. The main advantage of the algorithm is, that it is very small which means it does usually not need more than 400 lines of code.

In any suffix stripping algorithm for information retrieval work, two points should be considered when doing suffix stripping. Firstly, the suffixes are being removed simply to improve information retrieval performance, and not as a linguistic exercise. This means that it would not be at all obvious, under what circumstances a suffix should be removed, even if we could exactly determine the suffixes of a word by automatic means.

We already described the problems that can occur while stemming terms above when having discussed the stemming of the German language. Exceptions and problems are almost language independent. For that reason only the characteristics of the Porter Stemming Algorithm will be described here.

3.3.1 Introduction to the Porter Stemming Algorithm

The algorithm is based on suffix stripping. This may be done by removal of the various suffixes like -ED, -ING, -ION, IONS. By using this easy form of suffix stripping the terms: “connect”, “connected”, “connecting”, “connection” and “connections” all build the same stem which is “connect”. An side effect of that process is, that the stripping process reduces the total number of terms which reduces the size and complexity of the data in the system. Of course the success of this approach, with the use of suffix lists with various rules is always significantly less than 100% independent of how the evaluation takes place.

Although the presented algorithm is easy and short does not automatically mean that it is worse than more sophisticated programs. On the contrary, the presented procedure is even faster and more efficient than high cost alternatives. The exact procedure how the algorithm works can be seen at [PorterStemmer] and will be not explained in detail here as the common methodology is similar to the German stemming procedure.

Managing the Information Flood

4.1 Finding and Preparing Information

When asking the chief information and technology executives at most companies about what keeps them awake at night, the results might surprise you. They don't necessarily toss and turn thinking about virus attacks or even e-commerce initiatives. Their real nightmare is trying to manage an almost inconceivable flood of unstructured information, a flood that is expected to grow 200 percent per year into the foreseeable future, according to market research firm The Yankee Group.

When an employee wants to search information in the Inter- or Intranet, information about the company has to be found and collected. There are different possibilities how to find and locate information in an IT-System, whereby a combination of different methods can lead to good performances. If the fully automatic preparation is used, a program is following, similar like a user, hyperlinks on web-sites or it scans incrementally in directories on a server. Those programs are called robots or spiders and they can process all found documents fully automatic. While processing a document or webpage the program extracts relevant information like the name of the author, the title or some keywords. [Knögler]

Relevant information can be extracted from documents by using an algorithm which prefers frequently used keywords instead of seldom used words. There is also the possibility to pay more attention to words which are a headline or which are bolded or italic. Some documents have certain meta-tags which should be considered as well. Another possibility which is very time intensive is to process the whole information of the text with an a suitable algorithm. If a document management system is used meta-information about the document or explanations can be added by the user. Some systems force the user to input some meta-information. Other information-systems have an automatic search and indexing system. Whenever a document is checked in or out the search index will be updated. This has the advantage that the information is always up-to-date [Knögler 1999].

Basically spoken, companies differ internal and external information. To keep external information, especially when a huge amount of documents will be used, up-to-date is almost impossible. Mostly when using the world wide web as an information source anyone is depending on the particular search engine you being used. If one wants to manage the actuality and the availability of the information more self-responsible, she can administrate the links of external search engine sources on your own, which of course means a lot of administrative work for the company.

4.2 Weak Points in Detecting Information

It is important for the user who works with a knowledgebase, that the search engine offers its documents completely and newsworthy. While the view is restricted to an intranet here, an IT-system can handle this problem, but the search engines in the world wide web do not aspire this completeness as it would need too much effort. The enormous technical effort it would need to keep all documents up to date can not be managed even by the biggest search engines on the web, so that the operator companies try to offer a partial completeness. When looking at the structure of links in the www it can be concluded that a completeness of all documents and its links can never be reached, as there is never a central point of entry where to begin. [Knögler 1999]

The same considerations affect the actuality of already indexed documents which contexts have been modified. Without a document-management-system in use, the modified documents won't be recognized. If the collection of meta-tags circulates in a certain time period there is a possibility to recognize the change after a while. If the collection takes days or weeks there is no chance to find an new article. A chance of solution is to launch many File Spiders without fixed intervals and the frequently updating of folders or files which changed proportionally often in the past. With that technique the speed of the information updating can be adapted dynamically.

Another problem might be the deleting or moving of existing documents. The links to the search results do not have a valid reference anymore. There are different solutions against this weakness. One is a verification of a reference by the system itself, before offering a search result. In the prototype which arises out of this thesis the question answering system offers the user to delete so called *dead links* manually.

Information systems which consider the named problems need a powerful hardware. An big disadvantage of today's search engines is the multiple scanning of the internet by many services which causes a high traffic on the net. The updating of information by all kind of services is responsible that many files are being transferred over the net which are requested by engines and which cause a lot of costs by the web servers. Of course; the same problem occurs in the intranet of huge companies, due to the fact, that many companies use the internet for transportation and connecting other departments, too.

4.3 Quality of Information

The importance of quality and reliability of retrieved information rises potentially with the amount of documents in the intra- or internet. The rating of documents can be done in cooperation with the content offerer or by the user feedback. It is obvious that fully automatic information finding always has a lack of quality, as the content and the information can only be reviewed superficially. For instance if a keyword has a wrong meaning, it can only be detected by an user. To manage those problems an intelligent information algorithm has to be used. Developers of search engines recognized the fact and try to implement intelligent algorithms.[Knögler 1999].

It is also obvious that especially big companies are enormously effected by the named problems. The intranet-system of such a company has to take care of the administration of its documents and has to insure the consistency of the same sources. Without existence of meta-information, a search for an specific information in a knowledge base

of a big company would be almost impossible. On the other hand, hyperlinks which lead to lost information have to be checked as well as other quality features have to be renewed from time to time. For instance a document about the “computer society in our days” from 1980 might probably not be the right thing somebody searches for in 2004, although she is interested in the subject.

4.4 Structure of Data

One of the most important attitudes of an knowledge- or document management system is the ability to store information in a structured way. By doing so knowledge assets can be categorized easily which helps again when trying to identify knowledge.

4.5 Usability

A special aspect for the success of knowledge is the consideration of the usability when building a information system. A common desktop, ease of use, consistency of representation of the content as well as a fast access are basic conditions which have to be fulfilled, so that a system will be used.

4.6 Metadata

By means of meta data as well as rating by experts the quality of knowledge, stored in documents, can be improved evidentially. A special aspect in a huge and distributed company is the use of groups of meta data. In doing so the same information can be allocated to different quality criteria in different departments or areas.

To improve the quality of information or knowledge it should be obligating in every KM-system to add meta information to a new object.

4.7 Access Control

An important factor when distributing knowledge is the access-control, so that sensible data can not be seen from all employees or external clients which are connected to the intranet. Another advantage of an access control is, that the quality of data improves when using such a control-system as the quantity of information will be reduced and documents, which are being used in the same department show up first.

4.8 Multilanguage

For a company that domiciled in many countries the multilanguage of the knowledge management system and the information saved in it is important for the quality of information. Furthermore a user interface language has to be defined and rules if documents and information can be inserted in different languages.

4.9 Filtering Information

The use of filtering is another possibility to reduce the amount of results from search by using additional information. The search process can be more efficient if a meaningful

filtering system is in use. As soon as the amount of information or documents rises, filtering itself has become a critical condition. This is particularly true for knowledge workers, which are confronted with a flood of information every day, whenever they have to access internal or external resources like the world wide web.

An intelligent mechanism which is suitable for the intranet would be a system which saves individual profiles. These profiles take the user's long term interests into account. With that kind of profile the meaning of the information, which results from a search, can be measured and the documents can be restricted to essential information.

With that method the information-overhead of a specific query, which includes a lot of relevant words, can be avoided easily. It is important that the profiles can be changed, so that they are dynamically as the interests of the user and its function can change, too. With that method an IT-system would be flexible and could react to the users wishes and could favour the context and requirements of the specific categories of the knowledge worker. [Borghoff 1997]

The use of user profiles can be combined additionally with the conventional techniques of information finding. In doing so other components for the information filtering can be used which send simple queries with a higher speed. A perfect system should offer the possibility to search as a standard or to search as an advanced user with filtering.

Another interesting point is the automatic updating of the user profile while fulfilling activities. Such a system is called Intelligent Agent System and can be used in Workflow Systems. Those agents learn and optimize their own behaviour during the users activities. [Borghoff 1997]

In distributed companies we have, additionally to that, points above the aspect of local dependencies of information. For instance it is possible to capture meta-information where or in which location on earth the relevant document is stored and to which department it belongs. Or sometimes documents are only relevant in a certain environment. A document about the law of taxes is only useful in the country it belongs to. Another important meta information could be the group the file belongs to. All this could help to make the information finding more efficient.

If a document has a limitation by the range of validity and its receivers, the knowledge finding can be made fundamentally more efficient. In this case, documents can only be found by its defined departments and the persons who have access to it. A possible way to realize this is to use the meta-information of the document and to decide to which group the owner belongs. After that access can be restricted only to persons of the same group.

4.10 Improvements by the Prototype

The implemented prototype takes into account some of the discussed issues above. For instance the obligate input of meta data is being realised. Users have to enter a additional explanation to their question as well as a category to which it belongs to. Furthermore the date and the person who asked and answered the question are being stored with the question answer set. Out of it, one can contact these persons if he has a similar problem.

In the prototype there is no automatic detection if a link is a *broken* or *dead link*. But as soon as a user follows a *broken link* she can delete the link in the knowledge management system, so that the quality of data is being retained over the time. This has

the advantage that traffic on the net is kept very low, as no local files or web resources are being scanned automatically. This quality mix guarantees high performance on the one hand and a ease of use on the other.

the structure of data and the access rules collaborate in the prototype. Every new object or better every new question being asked in our question answering system has to have a certain category. Every resource that can be accessed for answering questions needs certain access rules for groups of users. Only web resources like Google do have global access rights. With the described mechanism two structure schemas are implemented. Data will be structured by Category and Access Right which makes it easier to administrate.

Another aspect of the prototype is its Multilanguage. The system supports German and English. An Administrator can decide if resources are being scanned in German or English. A feature that is not supported by many systems.

So far the question answering system disclaims any filtering, as a good filtering system would cost too much programming effort.

4.11 Conclusion

The problem of information overload concerns huge and geographical companies in many cases. On the one hand the information arising in the intranet of a company can quickly rise. On the other hand as documents are being stored on different servers in different places redundancies can occur. If documents exist in different languages the problem of redundancies becomes even worse as well as the information handling. Sometimes the meta-information usage varies from one department to another department. All these problems finally cause more information to be administrated.

Another source of external information we have not mentioned yet, are essential external resources like knowledge experts. If a company wants to use these experts, an individual strategy has to be made, or the company provides their own resources to administrate external resources. A well suited global player always has the possibility to acquire important external knowledge owners like experts or innovative enterprises.

5

Description of the Implemented Prototype

5.1 Description of the User Interaction Scenario

5.1.1 Introduction

In this section, the concrete scenario of a user query will be described in detail as a combination of the inclusion process of different internal and external information sources and on the other hand the building of a knowledge repository following the users interaction with the system.

The users view is simple, should contain a “Google” like user interface and a step-by-step “wizard” interaction with the system to evaluate the quality of the answers and possibly proceed from Step 1 up to Step 5 with increasing complexity until the problem is solved.

In a way this “wizard” like system could also be regarded as a *knowledge proxy*, that guides the users through various levels of interaction with different systems, starting from the simplest and cheapest approach (internal knowledge repository) and ending with the most expensive (query of other users, or external assistance).

5.1.2 Step 1: Answer given directly by Knowledge Database

The diagram in step one shows a typical case scenario at which a user’s query is being answered by the knowledge base (1.). After the initial installation and setup of the system, the knowledge base is empty. Over the time the knowledge base is filled with a set of questions and answers. Evidently, the larger the database the better is the expected hit ratio for an answer to a certain question. The user’s query will be processed by the question management engine (no other resources are involved in so far). This software module decomposes the question to a common form, so that every declension of a word has one common stem. This process is known as stemming (e.g., Braschler et.al 2003).

After decomposing the query the question management engine searches in its knowledge database for an appropriate answer (2.). The system evaluates matching answers by comparing the questions, which already have been answered, with the current question. If a certain similarity is given the database delivers the relevant data back to the engine which passes it to the user (3. and 4.). A similarity is given when words build the same stem as described in chapter 3. The grade of accordance has to be defined by the software.

Finally the user evaluates the answers delivered by the system. If the user appraises the answer to be a solution to his or her problem, the question (as long as it differs from other questions linked to the answer) will be added as a further possible question to this particular answer. For that reason the knowledge base grows although no direct knowledge is added. The users might also come to the conclusion, that the given answers were not sufficient. This would lead the system to step two, to refine the information retrieval process.

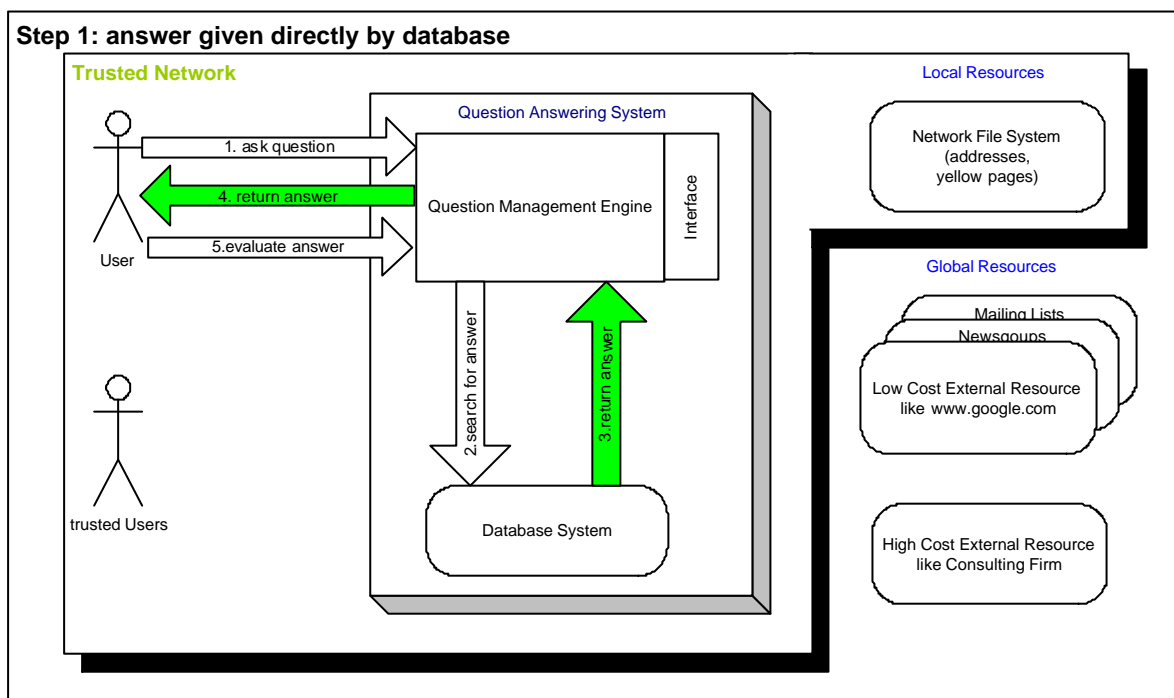


Image 5.1: Step1 answer process

5.1.3 Step 2: Answer given by Local Information Resource

The scenario is almost the same as in step one. By now the knowledge base offers no appropriate answer to the question. This means that either the overlapping percentage of the decomposed question, asked by the user, compared to the questions in the knowledge base is too low (2. and 3.), or the answer given by the question management system is rejected by the user. The procedure of rejecting an answer is not explicitly shown in the diagram as it would unnecessarily complicate the whole graphic. In our approach another software module, establishes a connection to other resources to find a good answer to the question (4.). (The function of this interface and its structure will be described later on.)

A local resource can be imagined as an extension of the knowledge-database system. Local resources can be arbitrary in type, e.g., databases, file systems, XML repositories and so on. However, the local resources should be catalogued or indexed before they can be used (at least, when they do not offer a powerful searching capability like a database system itself.). This catalogue content (index) has to be saved in the knowledge-database

system and has to be updated regularly as file or web pages change or move from time to time. From that point of view sequence 4. and 5. in the diagram just show the access to the local file system whereas the keywords for detecting the local resource are stored in the database system itself. But still the procedure of how to access local resources, which also could be another database, is a question of software design. The design is planned to be relatively open so that every programmer can implement a new module for the interface based on his own requirements.

Finally a set of answers or a link to a file will be returned to the question management engine (5.) which again passes it over to the user. The user evaluates the returned answer(s) by giving the system a positive or negative feedback. If an answer is satisfying the set of questions and the links to the relevant resources will be saved in the knowledge database. Periodically the system has to check if links to resources in the database are still valid since every dangling link deteriorates the usability of the whole question answering system. If the user sends back a negative feedback or if no answer could be found the question management engine has to proceed with step three.

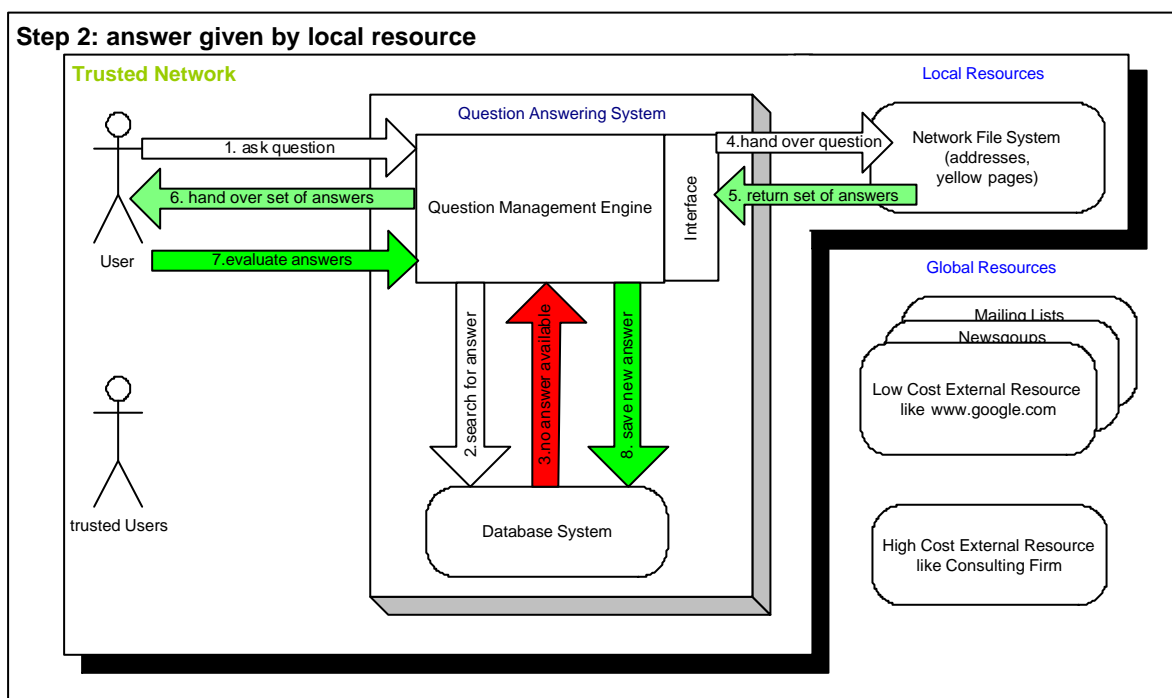


Image 5.2: Step2 answer process

5.1.4 Step 3: Answer given by a Global (External) Resource

The third step is very similar to the second step. Again the user formulates a query (1.) and the knowledge database has no appropriate answer (2. and 3.) or the answer is rejected by the user as described above. Now the interface-module gets into the game again. The scenario represented in step three shows the local resource as not available which could be caused by several reasons. The simplest reason is, that the interface-

module has not been implemented yet. Another could be that the user, asking the question, has no right or no access to the local resources being offered.

Anyhow, since the question answering engine cannot find an answer in its database the query is passed to the interface. The interface then “consults” the module which handles the global resources like search engines, newsgroups or mailing lists (4.). It is important to mention here, that new resources can be easily added: An interface has to be written (as described below) and this interface/resource has to be registered to the KM system.

As those global resources usually return a huge amount of answers (5.), it is the task of the interface-module to screen the best answers offered and to forward them to the question management engine. The engine sends the best answers back to the user (6.), who possibly makes an evaluation for one answer, which meets his expectations best (7.). In this positive case the question, the ranking and the link to the resource will be saved in the knowledge database. It is up to the interface-module if just the link to the resource or the whole resource will be saved for later use in the knowledge database. At the moment the link should sufficient.

Often a web page changes its content or moves to another place after a while, which makes it only temporarily available. As the database grows the checking of the links could absorb a lot of computing capacity. Hence a link to a web resource loses “attraction” while it has not been verified for a certain time.

If the user responds to the system that none of the answers being offered are solving his problems the question management engine goes on with step four.

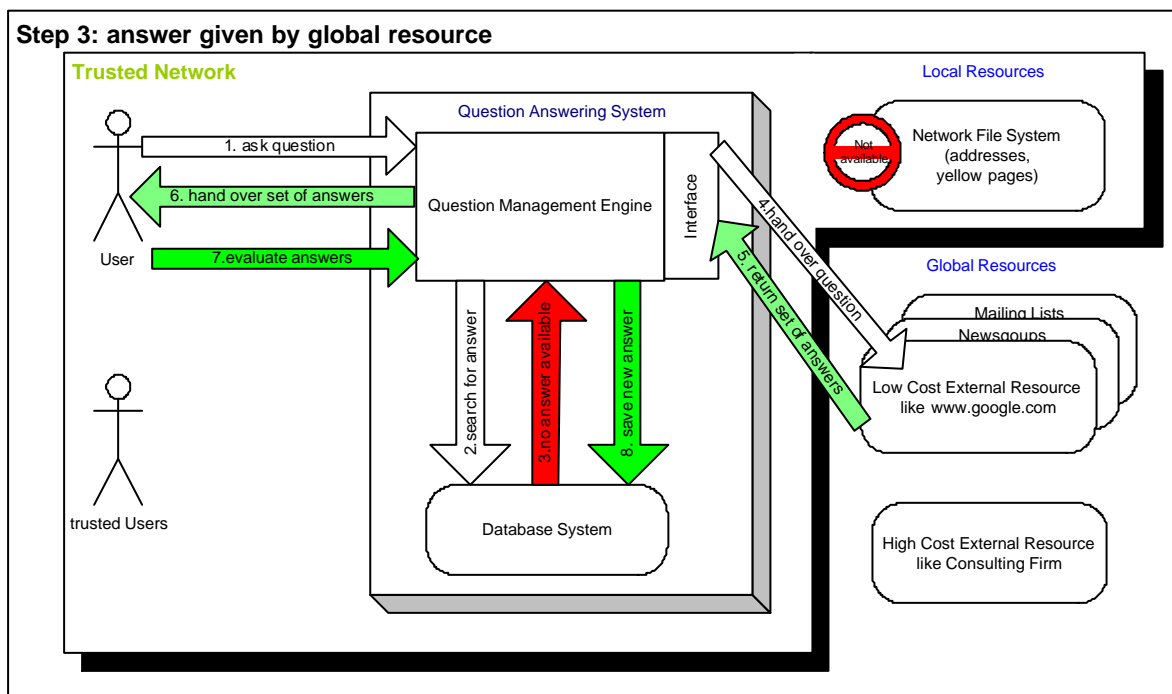


Image 5.3: Step3 answer process

5.1.5 Step 4: Answer given by Trusted User(s)

Step four in the question answering process might be of most interest of since for the first time another person or user will be involved in the answering process. First of all the user again formulates a query (1.), which is neither registered in the knowledge database (2. and 3.) nor answerable by the use of a global resource (4. and 5.). Obviously (as described in step two) the answer can not be found in the local resources either. So far all “cheap” resources have been exploited which forces the question management engine to transfer the question to human actors, e.g. a group of users being part of the project, the organisational unit or the company. Every authorised participant of the system has now the possibility to access these open questions.

As soon as another user adds a comment to a open question, provides an answer or puts in any kind of feedback, the question answering engine will return that feedback to the user(s), who had asked the original query(8.). Right now the user can enter into a dialog with the adequate person(s) by asking more details about the needed information or she is just satisfied and evaluates the answer as positive. It is also possible that multiple staff members take part in a discussion about an open question (even a notification that other users are interested in this problem is an important fact, as will be noted later). As soon as more comments are being added to the open question every user taking part in the discussion will be notified that a new comment was added to the open question until the problem is solved and the question will be closed.

Once a question is forwarded to other staff members the user asking the question has to attach points from his score account to it. Depending on the difficulty and his personal interest he can add high or low scores. The score system will be described in details later. If after a certain time the question will not be answered by any staff member and is still marked as open in the system, the question management engine proceeds with step five.

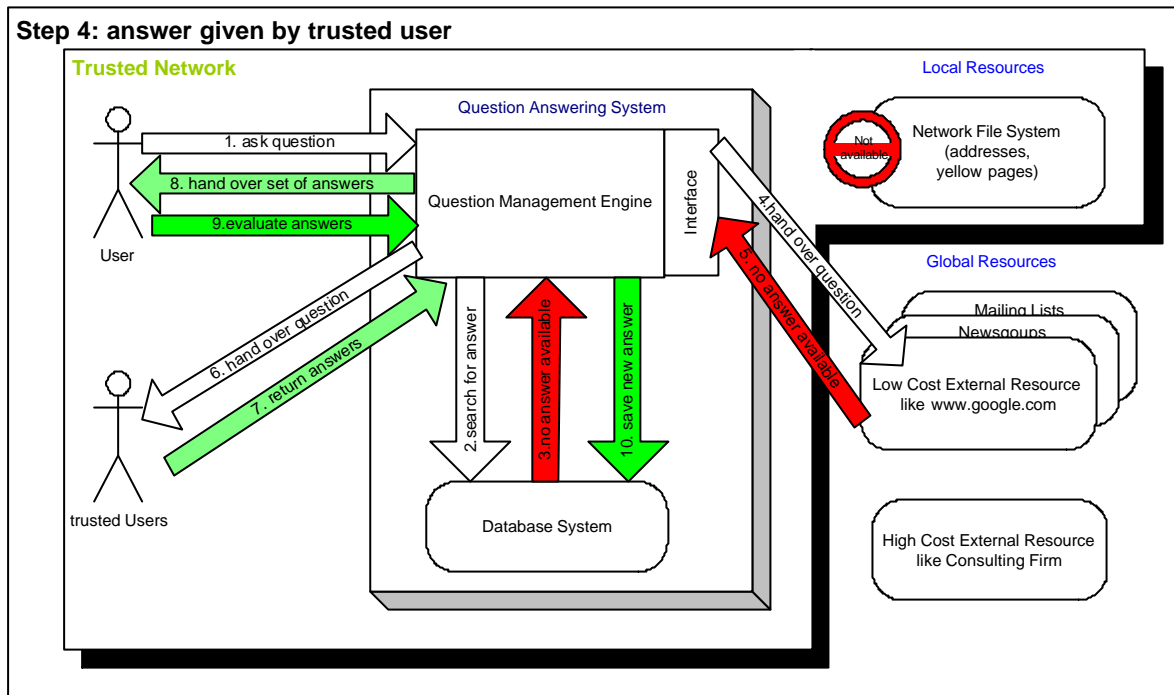


Image 5.4: Step4 answer process

5.1.6 Step 5: Management Activities

If all the steps above did not lead to a solution of the open query, this particular problem becomes a management issue. Now the (project) manager or team leader has the responsibility to evaluate the severity of the problem. This process is aided by the points attached to a problem as well as by the number of users that marked this problem as “interesting”. Different solution strategies can be imagined, starting from explicit order to internal staff to solve this very problem, up to the usage of external consultants. Those further steps have to be decided by management and cannot be automated by obvious reasons.

Eventually the answer to the problem has to be entered manually afterwards, to save the question and answer set in the knowledge base. If the problem is too complex to be described textually it might be a help for the questioners to find another staff member who lastly solved the problem. The questioner can then contact this person to get some help from him or her. This leads to better networking and efficient communication. This process of the last step goes in detail:

The user starts by asking a question to the question management system (1.). After searching in the knowledge database for an adequate answer the engine passes the query to the trusted users (4.). These users try now to help the questioner (5.). If after a certain time period no fitting solution could be found the question management engine finally contacts the (project) manager to eventually consult a high cost external resource like a consulting company (6.) which is specialized on the problem. Ideally the “external help”

now solved this particular problem (7.). The solution is handed over to the user (8.), who evaluates it (9.). After that the solution and the question are stored in the knowledge database (10.)

As described above the software-interface and its programmable modules are open. The described steps here are just a default suggestion which makes sense for a normal scenario. But the order when to consult external, internal resources or staff members should be freely selectable in the implementation process and highly depends on the structure of the company or the project. For instance it might be unbearable for a project which has a completion date of one month to wait two weeks for an answer. For that reason the software prototype is designed in that way that the order of the different modules can be swapped easily.

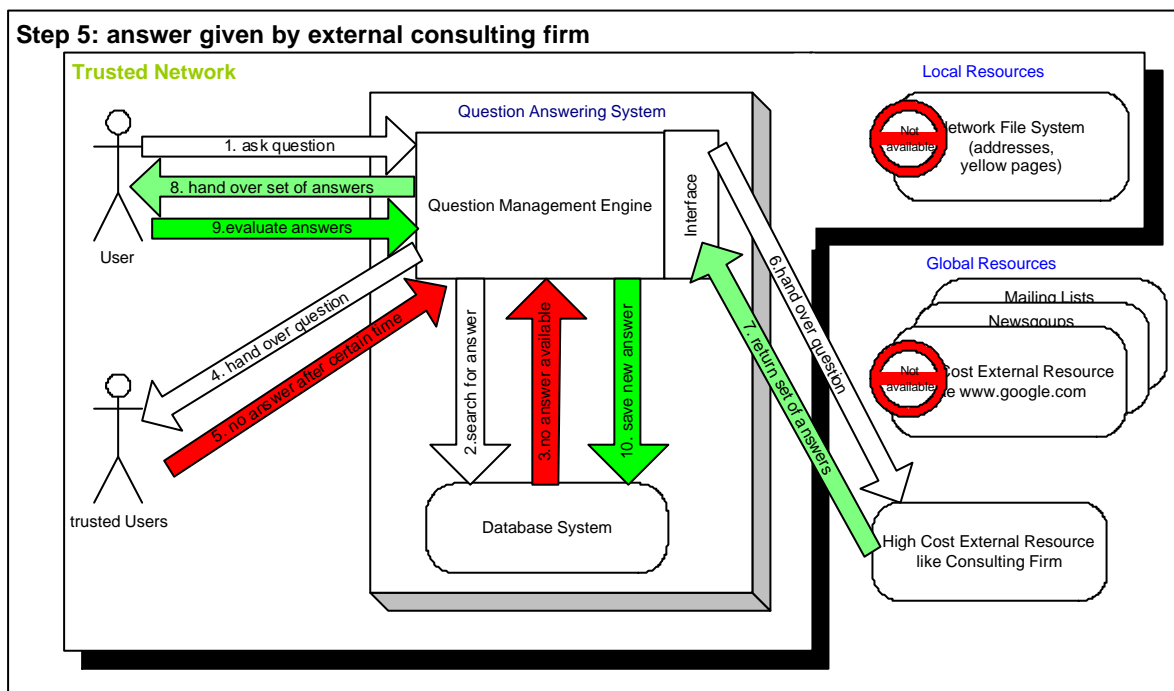


Image 5.5: Step5 answer process

5.1.7 The Scoring System

The scoring system should be the main motivation factor to keep the “trading” of questions and answers alive. Every new registered member of the question answering community receives periodically a certain amount of points (which possibly can be regarded as “virtual money”). These points can be used to rank questions. The more important or difficult a question is the more points from the own account can be added. But not only own questions can be donated. If a user detects a problem posed by others, he or she may add points from the own account to demonstrate the importance of the problem.

If another user helps the questioner to find a solution to his problem, she should be obliged to give him the points. If more users took part in the answering process the questioner can also decide to split her points up to more users or even increase them if he appreciates their helpful work. As soon as her account of points approaches to zero, she should be motivated to answer questions from other users as well. The fact that questions have to be scored and that points are used are economically good causes for the users to deliver cogitated questions and reasonable contributions.

Using statistical reports, managers can trace out which staff members have special skills, take part in the system and where there is need for more training for individual workers. Moreover people can be detected and brought together who have the skills that are needed in a special project. To give the score system a positional value the acquired points should be changeable in a kind of swap market. For instance one hundred points could be used for a bonus or a day off. The ideas of how to use points in another way than spending them for questions should not be given any limitations. In addition there should be a periodical list of the members with the highest score. To increase the amount of points managers or superiors can contribute additional points for employees as an incentive.

5.1.8 Sequence Diagram

The following diagram will show the whole process in a sequence diagram

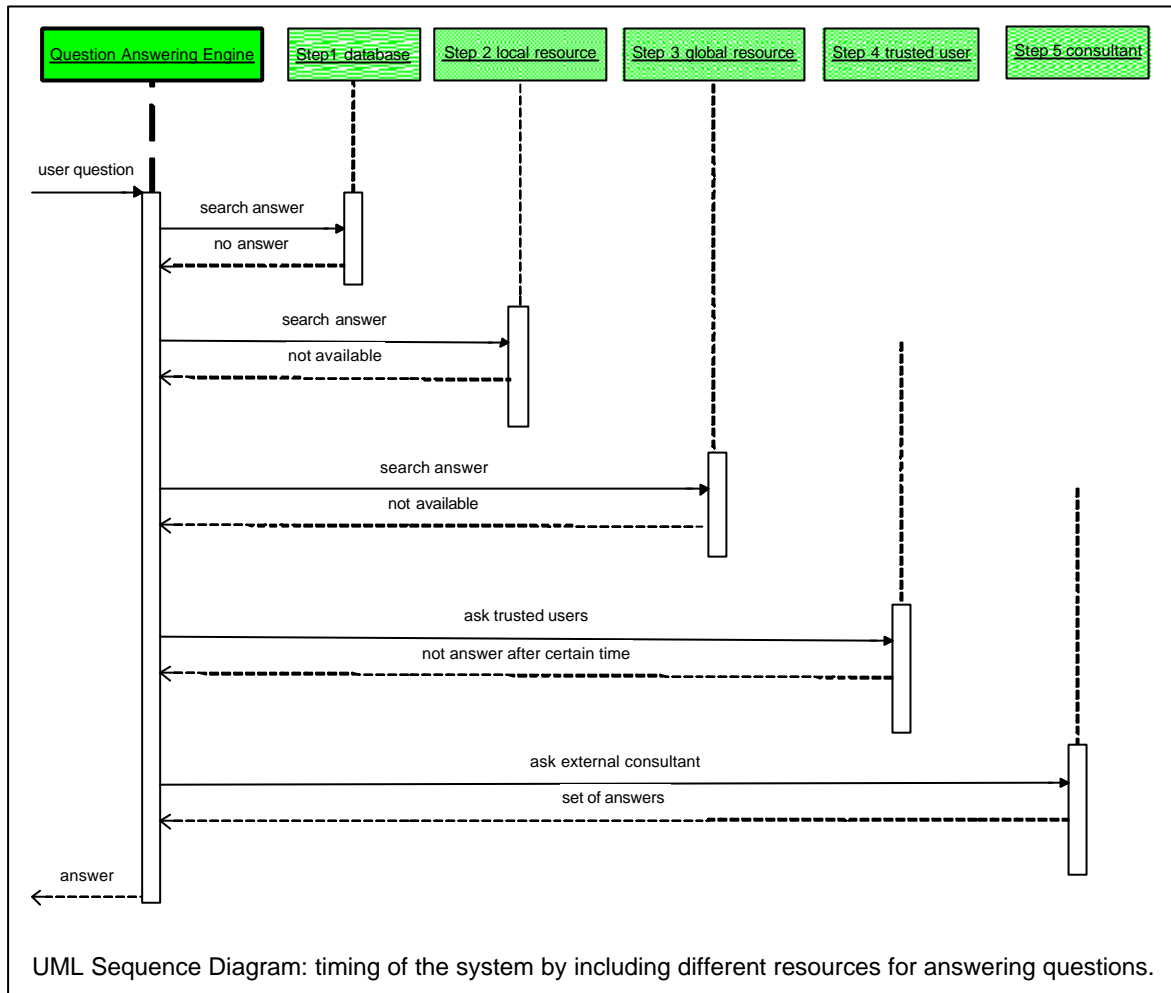


Image 5.6: Sequence diagram

5.1.9 System Integration – Technical Aspects

In this section the details of the software interface modules will be explained in more detail as it plays an important role in the whole system. The term “*interface*” used in this paper should not be confused with the *interface* term used in the programming language Java which is part of the heritage strategy.

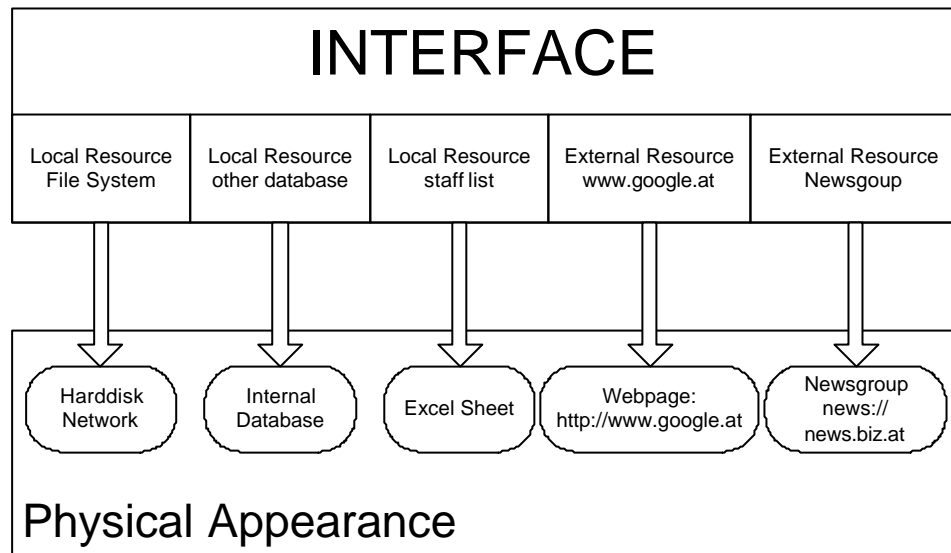


Image 5.7: Interface

The interface module in the question answering system is just a solid software module upon which individual modules can be hooked-up to expand the functionality of the whole system. One of this modules could be the interface to the local resources. Another could be the interface to the global resources, etc.. The graphic shows the interface with its modules and the particular physical appearance of the information sources. The solid interface sets up the guidelines for the individual modules being implemented. It is important to find the right policy mix of routines which have to be implemented and those given by the solid interface. On the one hand the system should be relatively open and on the other hand the solid interface should represent a stable basis on which other freely selectable modules can be linked without a big effort. Some modules like local resources need to pre-scan their resources frequently while others like web search engines cannot do that (as it makes no sense to scan the whole web). Only after locating a web resource and saving the link in the knowledge database one can verify the resource for its availability. This two simple examples show clearly that the complexity, of the different requirements, needs to be carefully planed and will always need some compromises.

So at least two types on interface specifications will be defined: One for resources that need to be indexed (like file systems, XML documents), and one for resources, that can (or should) be queried directly (like WWW search engines, relational database systems). Essentially the interface specification describes methods that forward the query to the specific subsystem, and a method to return the result in a unified way.

If a new resource needs to be added to the system, a programmer has to implement (in the easiest case) those two methods for the specific information subsystem, add some meta-information about the system and register it to the KM main-application. With the next user queries, this new registered service will be included into the information pool.

5.1.10 Motivation and Cost Saving Factors

“Employees often do not have time to input or search for knowledge, do not want to give away their knowledge, and do not want to reuse someone else’s knowledge.” [Russ 2002].

Participation of knowledge owners and future users is an important factor for the success of knowledge management systems. Many knowledge management systems failed, because of their lack of participation. Knowledge owners did not have the time or the intention to write down their skills. In many departments one person is the specialist of a particular domain. But that person is most probably the busiest and therefore the bottleneck of information flow. If she leaves the firm all her skills will accompany her. Especially these staff members often do not have time to do proper documentation of processes. While they keep on developing software for instance, the amount of undocumented software grows.

Finally one does not know where to start as the time for the whole documentation is not available. By choice people might not do any documentation at all, since there is no direct benefit of doing so. Exactly at this point a question answering system can be the solution. The direct benefit is given as someone helps solving a problem. It is a motivating process to see that one’s skills are needed and it usually does not take more than five minutes to answer a question. The problem of documenting a huge amount of processes is split up into little pieces which makes it more convenient for knowledge owners to participate. And even more importantly, the process is problem driven, that means, in this approach solely the kind of knowledge is documented that is *really* needed! This is an important time saving factor as traditionally documents have been written that will never be used again. A software developer can now decide if it is necessary to describe a routine or an operation more precisely if other team members permanently press him for help concerning this routine. In that case she writes an tutorial on how to make use of an operation and sends the link as an answer to the users asking for help.

In many companies the problem exists, that staff members are dissatisfied with business procedures. They complain that procedures are old fashioned, long winded and untouchable in their execution. So it often happens that a question for improvement does not reach the responsible person and will be forgotten or lost -a frustrating and little motivating experience. Moreover, it is often overseen, that many users suffer under the same problems! These effects could be leveraged by using a question answering system based on a knowledge base. Questions or suggestions for improvements can be watched by the managers and heads of departments in a “democratic” way. After figuring out a problem or innovation suggestion, suitable countermeasures have to be launched. The systems should be able to act like an early detection system and should discover trends and grievances in a company.

Particularly the motivation of the employees and the idea that their help, innovations, proposals and complaints are being heard is a main reason to justify the use of a question answering system like this. When thought further on, such a system leads to better quality management and faster innovations. By now problems are documented and especially managers can take their time to have a look at it. Very often an employee meets her superior on the floor and confronts him with a problem. The superior on the other hand is on his way to an important meeting and has absolutely no understanding for the employee's needs, as he is too busy with his own matters at this moment. The need for a mediation architecture is obvious and the suggested system can be seen as such.

And finally, as mentioned already, the proposed system works like a information/knowledge proxy. And when designed and implemented properly, using the system will increase productivity and lower costs, as (1) employees should find solutions for problems faster (2) they will use available (expensive) information resources only when really needed and (3) transparency in the problem/solving process is added, hence traceability is increased dramatically (4) communication between employees is encouraged. Especially also the last point (not yet analysed in detail here) can increase the productivity dramatically, as Abecker et. al. points out:

“Coordination and collaboration support must be a first order citizen of KM [...] information retrieval and management systems must deeply be interwoven with the collaboration-oriented everyday work.” [Abecker 1999]

5.1.11 Access Control

Another important factor is the access control of resources. As web resources are available for every user group, do file resource have to be restricted to a certain group. Imagine that a secret document, which has helped to answer a question is access able for everybody. This could harm a company enormously. For that a access control is implemented in the prototype which allows an administrator to distribute rights to user groups. Before resources from the file system are inserted in the knowledge management system the administrator has to give authority to the users.

5.2 Introduction to Lucene

Lucene is a high-performance, full-featured text search engine written entirely in Java. Like all of the Jakarta projects, Lucene is maintained by a group of dedicated volunteers. It is a technology suitable for nearly any application that requires full-text search, especially the programming in Java makes it platform independent and furthermore perfectly usable for the our prototype. Lucene started by Doug Cutting as an independent project and around September 2001 it became an official Jakarta project.

A main advantage of lucene is, that it support both German and English, which is an important factor for our prototype. Lucene is free software and is governed by the Apache Software License (ASL) and thus it is free.

Lucene was designed quite open, so that it is not a big effort to implement interfaces and code of his own.

Lucene has scaleable high performance indexing system which indexes about 200MB of raw text per hour whereby the index is about 30% of the size of the original texts.

Moreover it needs little system resources and has a powerful search algorithm. Search results will be ranked by the hit ratio, which means that the best result is offered first. While indexing a text file one can subdivide a document in different fields like author, content, date etc. Afterwards it is possible to search exclusively in that fields. It should be mentioned that it is up to the design of the prototype which documents or which fields of a document have to be indexed, as one should not forget that a huge amount of documents need a big harddisk for the indexing as well. Sometimes maybe the abstract or simply the filename could be enough to retrieve the information again. It is not always necessary to index the whole content of a file.

Indexing is the process of creating the 'index'. The index is a special database that contains a compiled version of the documents and is optimized for quick lookup for a list of documents that contain certain words. These words sometimes are also called terms.

Lucene API provides exact control over the information stored in the index for each document and how this information is used during indexing and searching. On one extreme, it is possible to store just the location for each document, and index the content of the document as a monolithic piece of text. On the other extreme, one can store the entire document as well as various attributes such as Author, Title, and Date and perform searches that consider these attributes for matching and ranking. Typically, the index is stored in a set of files that Lucene creates in a directory of one's choice. It is also possible to save the index in any database.

It is pretty easy to index a directory of text files. All one has to do is to create an instance of `IndexWriter()` and then iterate over the documents in the directory. For each file one has to create a Lucene Document object and add it to the `IndexWriter`. The `IndexWriter` must be given an algorithm how to tokenize the words found in the document. As mentioned before this could be an algorithm to stem words in German or English. The used algorithm is also responsible for deleting all stop words like "a", "the", "and" etc. as they are of no use. The following example should explain how a simple indexing process will be implemented.

```

class IndexFiles {
public static void main(String[] args) {
try {
    Date start = new Date();

    IndexWriter writer = new IndexWriter("index", new StandardAnalyzer(), true);
    indexDocs(writer, new File(args[0]));

    writer.optimize();
    writer.close();

    Date end = new Date();

    System.out.print(end.getTime() - start.getTime());
    System.out.println(" total milliseconds");

} catch (Exception e) {
    System.out.println(" caught a " + e.getClass() +
        "\n with message: " + e.getMessage());
}
}

public static void indexDocs(IndexWriter writer, File file) throws Exception {
    if (file.isDirectory())
    {
        String[] files = file.list();
        for (int i = 0; i < files.length; i++)
            indexDocs(writer, new File(file, files[i]));
    }
    else
    {
        System.out.println("adding " + file);
        writer.addDocument(FileDocument.Document(file));
    }
}
}

```

Creates a new index instance in the directory „index“ with a StandardAnalyzer

Jumps down and iterates over all files if a directory is passed as parameter.

The object FileDocument subdivides the document in different fields. (title,content,date)

Adds the document to the writer object which indexes it.

Source code example: how to index a directory

Image 5.8: Example1 Lucene

If one wants to update the index, because some documents change it is possible to either re-index all files or just delete and re-index the documents which have been changed. With the `IndexReader.delete()` method one can delete an document from the index. As shown in the example above a `FileDocument` object when calling the `IndexWriter.add` method has to be passed. The constructor of The `FileDocument` class subdivides the document in different fields and returns the subdivided document object. This class is shown in the following example.

The document is subdivided in three fields:
 path: for the path of the document
 modified: which includes the last modification
 content: which is the whole content

```

public class FileDocument {

    public static Document Document(File f)
        throws java.io.FileNotFoundException {

        Document doc = new Document();

        doc.add(Field.Text("path", f.getPath()));

        doc.add(Field.Keyword("modified", DateField.timeToString(f.lastModified())));

        FileInputStream is = new FileInputStream(f);
        Reader reader = new BufferedReader(new InputStreamReader(is));
        doc.add(Field.Text("contents", reader));

        return doc;
    }

    private FileDocument() {}
}

```

Add the path of the file as a field named "path". Use a Text field, so that the index stores the path, and so that the path is searchable

Add the last modified date of the file a field named "modified". Use a Keyword field, so that it's searchable, but so that no attempt is made to tokenize the field into words.

Add the contents of the file a field named "contents". Use a Text field, specifying a Reader, so that the text of the file is tokenized.

Source code example: how to subdivide a document into fields

Image 5.9: Example2 Lucene

While we have seen in the last example how to subdivide a document in different fields, each field can have a status how it should be stored for later use. Generally each field can have three attributes which can be combined.

The three attributes are:

- **isIndexed** when true indicates that the content of the field may be used during searches to locate desired documents.
- **isStored**: when true indicates that the content of the field is stored in the index such that a complete copy of the value is available when retrieving the hits. This is useful for example for an URL of a document or a changing web content.
- **isTokenized**: applicable to indexed fields and indicates if the content of the field is broken into terms (words) or used as a single indexing term. For the prototype this means that a document will be stemmed and stop words will be removed.

It is important to make a right decision on how the different fields in a document should be stored. For example, if you want to include in your hit list the document title then set the title field to be stored. If you want to save disk space or if your documents are huge, set your document content field to be tokenized and indexed but not stored. If you want

to keep the document modification date, set the modification time field to stored but not indexed, unless you want to search for documents of a specific date.

So far Lucene does not support the analysis of documents like PDF's from Adobe or Microsoft Word or Excel. Only documents of the ASCII format can be indexed or searched by Lucene. But there are solutions which make it possible to convert documents in the named format in that way that lucene can handle them [lucene converter].

In the Lucene description it is referred to an analyzer which simply a stemming algorithm and a stop word remover as it was described before. For the prototype two analyzers are used. One for the English language and the other for the German. One of the fastest and easiest analyzer in English is the Porter Stemmer which is implemented in Lucene. One has to remember that the Porter Stemmer is only usable for the English language and may give an unpredictable result for other languages [Porter 1999]. It is important to know to use the same analyzer for indexing and searching as the question will be analyzed the same way the index has been created. A different analyzer object for indexing and searching, likely returns no or some wrong results.

While so far we were dealing with the indexing process we now shortly have a look at how Lucene searches for documents being indexed before. Searching with Lucene is the operation of locating a subset of the documents that contain the desired content or their attributes match some specification. The input for a search operation is a 'query' that specifies a criteria for selecting the documents and its output is a list of documents that matched that criteria. The output is given a a hit list, where the best result is the first in the list.

The search operation is performed on the 'index' which is a specialized data base that contains a pre compiled information of the document set as mentioned above. The index data base is optimized for locating documents, which contain certain words or terms, quickly. The index data base is being created during the 'indexing' process as it was explained before.

In general, a query is a specification of the content and the properties of the desired documents. Every search is done by matching a query against the document index and locating the ones that match the query.

A Lucene query is represented by an instance of the base class Query.

The simplest query specifies a single word which is called a term, that is to be matched against a single field of each of the documents in the index. This kind of query matches any document that contains the term in the specified field. As described before every document has different fields: For instance title, author or date. More complex queries may now contain more words, which are connected trough "and" or "or". Those words are called Keywords. Some queries can also match only against a special field of an document, for instance it makes sense to find documents not older than ten days in the date field and not in the title field.

A special attribute while searching for content is the boost factor. When searching for more than one word, a boost factor can be added to every word, meaning that the occurrence of that word will be ranked higher or lower in the search result. Boost factors are useful when the query contains several terms, possibly for different document fields, and it is desired to boost the scoring of document that contains specific terms. A boost factor must be a number greater than 0.0 and is by default 1. The following example shows an easy way how to search for a term in the indexed database.

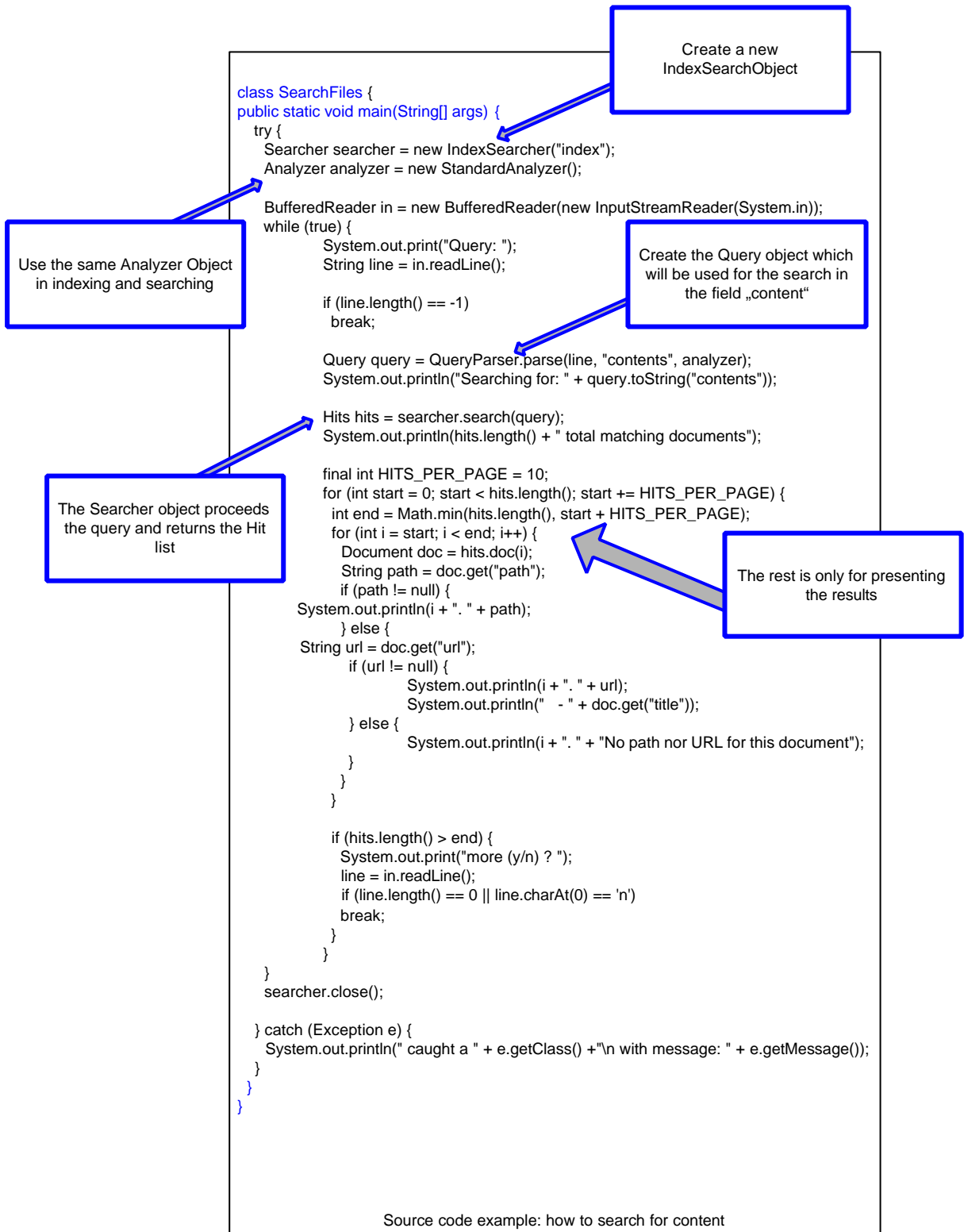


Image 5.9: Example 3 Lucene

It is possible to have a full functional system with only these three source code examples. To use Lucene in the question answering system a lot of sophisticated changes have to be considered. This will be explain in detail later.

Lucene describes an easy way to add a security schema to the returned files, so that some users do not have access to all files. A simple way to protect documents is to perform the search, and then display only hits that match the security criteria. A more efficient approach is to integrate the security filtering in the query itself. This can be done by adding clauses to the query, which will make sure, that only documents are visible to the current user, which are returned in the hit list. One can do that using negative clauses that will eliminate not desired documents or positive clauses that will enable desired documents. One just needs to make sure, that the document information in the index has enough information to determine if it is visible to a specific user or not.

A third approach is a combination of the two. Restrict the search by adding the necessary clauses and then run an addition security check on the returned hit list, just before displaying it to the user. This way it is possible to have both, the performance benefit of the second approach and the extra assurance that information is not leaked to unauthorized users.

The combination approach is especially useful when it is impossible to have clauses that will restrict the hit list to exactly the subset of authorized documents. In this case, you restrict the hit list to a super set of the authorized documents (for extra performance) and then perform the final security filtering on the returned hit list.

5.3 Introduction to the Prototype of the Question Answering System

5.3.1 The Question-Based Approach

One of the major goals for the question answering system is to offer a user with even little computer knowledge a desktop where she can find her way to information that might come from different heterogeneous sources. Every user with little IT knowledge has his own methods to access information needed. Some scan the whole hard disk or the network for a file, while others make copies of every file they need. Even by choosing the right web search engine different preferences can be detected. And many users do not even know how to find all relevant information sources! Hence to receive the information needed one usually has to contact different sources.

In a working environment often quite similar problems occur over and over again in such a way that a single coherent system with only one user interface might enable to solve these problems more easily and moreover simplify the search process. Considering such an “information portal”, that centralizes the information retrieval activities of all users, there is an important “side effect”: As questions are posed through one central system and answers are collected by this system, those question/answer activities can be analysed and processed by this application. As a result those activities can be used to build up knowledge that will be saved in a way that it can be accessed again. The proposed system does not only delegate queries to other subsystems and collect the answers, but in case that this procedure does not lead to a success (in terms of solving the users problem), also stores the open questions which are in the system and encourages staff members to answer/solve open issues. To increase the workers’ motivation and to

ensure the quality of the knowledge base, a scoring system will be implemented. This complex mechanism will be described in detail below.

As mentioned earlier, a meaningful knowledge management software must be embedded in a worker's everyday practice. To attain a tool that will be used as the "standard information finder" a user must find her daily needs of information with "one click". It should be the top priority to make it comprehensible to the user, that using this system is the most efficient way to obtain information from different sources.

The prototype of this question answering system is being implemented in a client/server architecture with a web-browser based user interface and a Java application server backend. So it is possible that every user has its own information portal.

5.3.2 The User Interface

The user-interfaces is embedded in the OSWP Open Science Workplace web GUI. The frames are generated by the Struts Framework called Tiles. As soon as the user clicks on the KNOWLEDGE button on the top bar the following site will be presented.

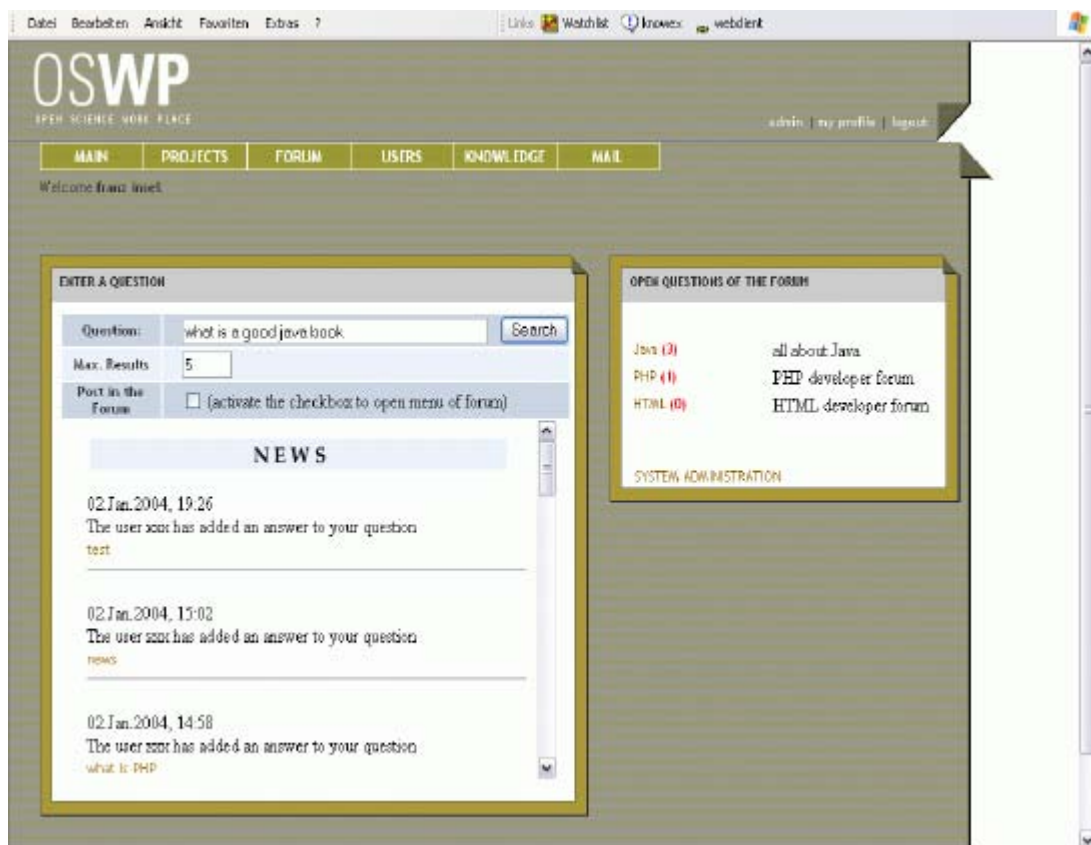


Image 5.10: Userinterface Mainpage

The website has two frames the left and the right one. The right frame shows all categories, which can be manipulated in the administration area, and a description to it. The red number behind those categories signifies the open questions in this forum. A link

on each category leads to the forum. If a user has administration rights the link “SYSTEM ADMINISTRATION” will be shown which leads to the administration area. The left frame is responsible for the search process. A user can add an question in the

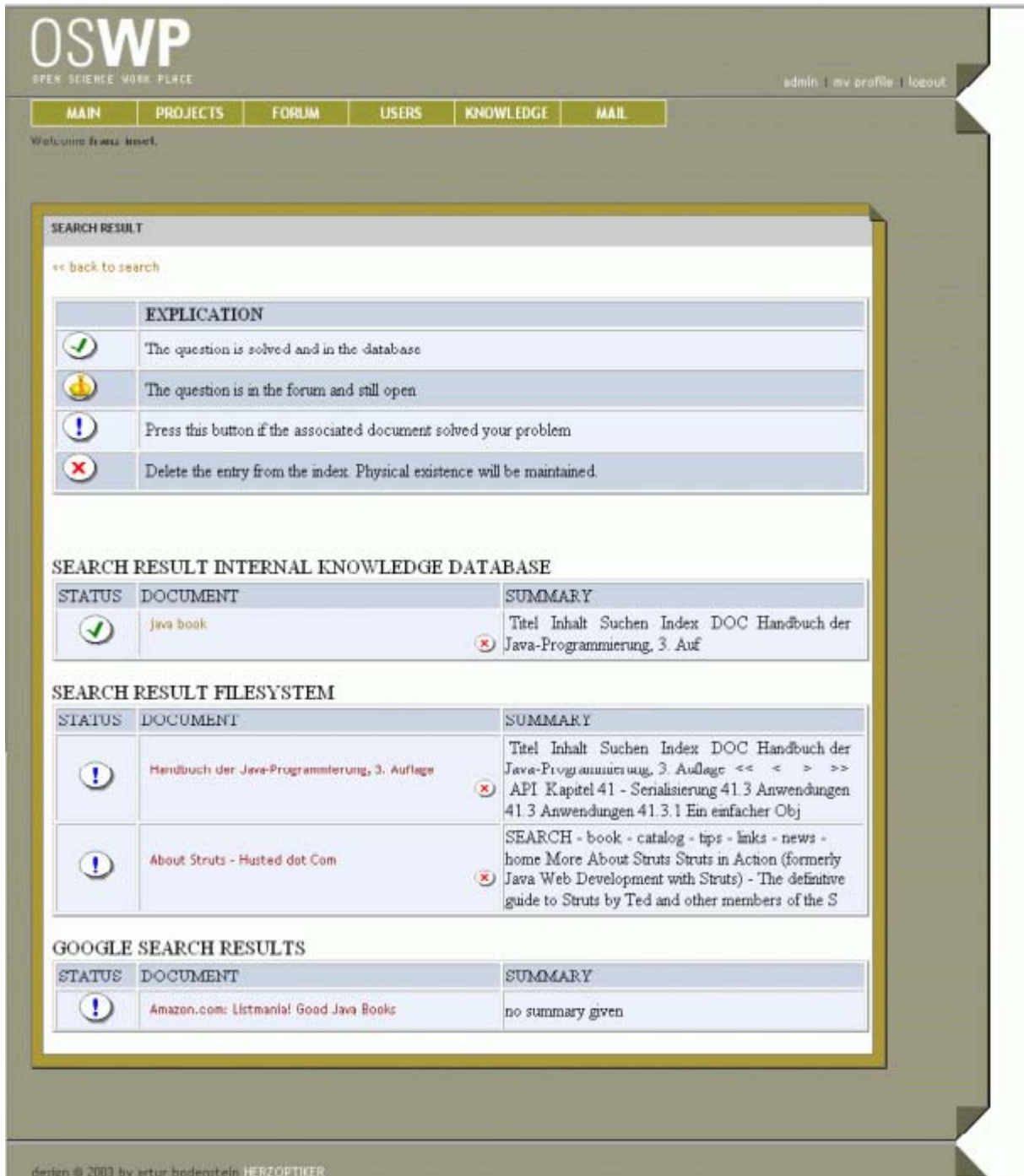




Image 5.11: Userinterface Result Page

text field and press the search button to find a suitable answer. Right now the user asks the question:” what is a good java book” and desires not more than 5 results. The field

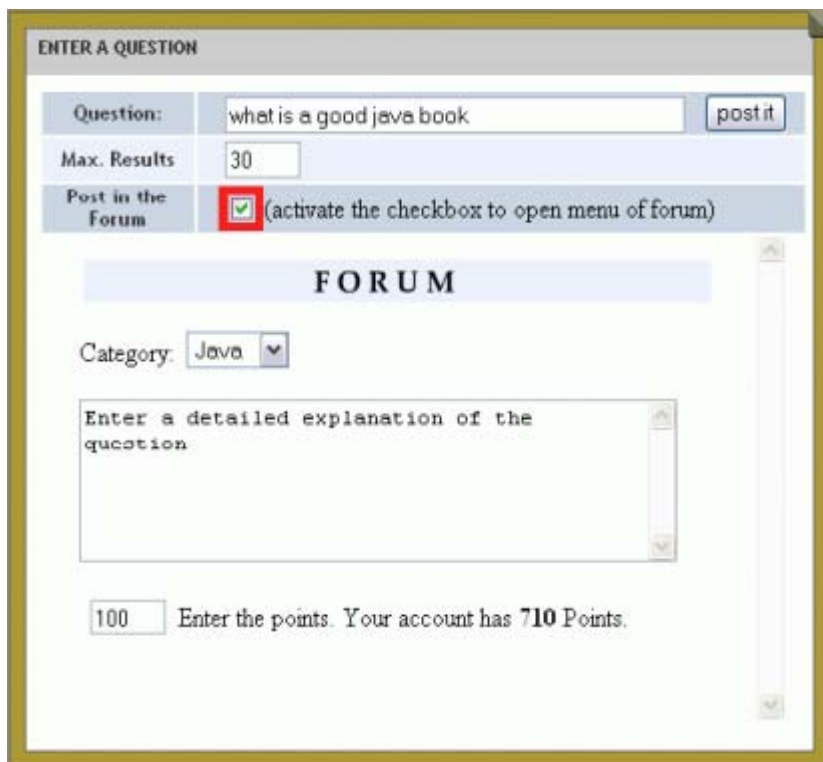
max. Results holds the number of the desired maximum results given by the question answering system.

The news area below gives individual information for the logged in user. As an example the user can see here, if someone answered to his questions, if a user confirmed a question in a forum he took part and so on.

If the user submits her search query a typical page that shows up would look like Image 5.11.

The result response page of the question: "what is a good java book" has four answers. On top of every search result page a explication of the symbols is shown in a table. The symbol  signifies that a question has been either in the forum and is now answered or was a normal question from a user which has found a document or other resource which solved his problem. Only questions that have been answered and are now in the internal knowledge database can have this symbol. If an question has been asked and put in the forum the dollar bag appears . Every time a user wants to put a question in the forum he is obliged to give a certain amount of points of his own account to encourage other users to answer his question. The dollar bag shows that someone still can earn points by answering this question. On the other hand it makes the questioner clear, that someone else has a similar problem which has not been answered so far. It gives him the opportunity to take part in this discussion to solve his own problem. This symbol can only be shown in the internal knowledge database results. In the picture above no open question in the forum is available.

Sometimes external resources like files moved to another place or have been deleted. The same may happen to





references to resources on the world wide web. As most search results are taken from an index it can happen that links are not valid anymore. To keep information up to date and to improve usability for other users everyone can remove dead links in the index if he finds one by clicking the -symbol. After that the link will be removed from the index. The physical existence of the

Image 5.12: Userinterface Forum

document will be available, which means that the document will only be deleted from the index.

The questioner can now scan through all received documents. If one of these documents solved his question he has the possibility to enter the answer in the internal knowledge database by clicking on the -symbol. After that step, the question-answer-pair is stored in the knowledge database. Another user who will ask a similar question, receives the fitting answer on the top of his search result page.

If, for instance, the user did not find a document which represents a suitable answer to his question, the next step he can do is to post his question in the forum. So every member of the community has the chance to participate and to help him. The next image shows the main page of the knowledge system again. It represents the same status as before with the difference that by now the checkbox for posting a question in the forum is activated. (marked by the red box). The news frame disappears and the FORUM frame comes into the front. The questioner can choose a category which belongs to his question. In the text field below it a detailed explanation of the problem should be entered. Finally the user adds the amount of points he wants to give for a good answer. To submit the question to the forum the questioner simply has to press the “post it” button. The question is now

marked as an open question in the forum and can be seen by everybody as long as it will be closed.

The next image shows the appearance of the question in the forum. Other users can take part in the discussion and post their answers. The bag with the dollar sign stand for the points the questioner offers for a concrete



Image 5.13: Forum

answer. Whenever one of the answerers posts a correct answer, which solves the problem, the

questioner can transfer the points from his account to the answerers account. That stimulates the question-answer process.

The exemplary illustrations above are the most important parts of the user interface of this system, but it also seems important to explain the use of the Indexing-mechanism shortly. The image on the following page shows the File-system Indexing in the administration area.

On the top of the page, the index directories are listed. Below these indices one can find two functions: update and delete. The update should be done from an administrator from time to time. For instance, if many files in an indexed directory have been deleted or moved, the index is incomplete or inconsistent. The result for the user could be that he receives answers as links for his questions which do not exist anymore. A regularly update could avoid this effect. A scheduler, which updates all files from time to time is not implemented in the prototype. But this could be a meaningful addition in the future. Still the user has the possibility to delete *dead links* so that they will not be shown any longer.

If a directory has been deleted or if the administrator does not want to use the index anymore he can simply delete it.

To create a new index one can go to the bottom of the page and enter a directory to which the server has access. This directory should include some files of interest which are in a .txt, .java, .htm, .html, .pdf or any other supported format. In the second text field one can enter the directory where the index should be stored to, usually the directory will

The screenshot shows a web interface titled "ADMINISTRATION FILESYSTEM". At the top left, there is a link "<< back to menu". Below this is a table with three rows:

new index directory (will be created)	c:/x/index1
directory with textfiles (*.html,*.pdf,*.txt ...)	c:/x/texte1
Stemming Analyzer used	English

Below the table, there are two links: "Update" and "Delete".

Underneath, there is a section titled "Create another Index" with a form containing:

- new index directory (will be created): [empty text field]
- directory with textfiles (*.html,*.pdf,*.txt ...): [empty text field]
- Radio buttons for language selection:
 - german texts and files
 - english texts and files
 - english technical files

At the bottom of the form is a "Create" button.

Image 5.14: Administration Area Filesystem

be created automatically. To understand how the system works, it is essential to understand how the indexing works. The indexing scheme will be explained in detail in the following section. By pressing the create button the new index will be created. This can take a while depending on the on the size and amount of documents in the directories.

5.3.3 Indexing and Searching in Detail

As mentioned above the indexing of files can be done through the administration area. Every index can actually be in English, German or technical English. For instance if the index will be in English all the text files found in the directory and subdirectories are being analyzed with the Porter Stemming algorithm. Later on, when questions are asked, the questions will be analyzed with the same analyzer as the indexed directory. This idea makes it possible to index the same directory twice. One time in English and the other time in German. This could make sense if in the same directory is a mixture of English and German files. But one has to take care, because this leads to the effect that the same document could be shown twice as an answer to one question.

After the indexing is done, the user can start asking questions. The following image makes clear how the search process is being done internally.

In that example the normal user asks the question: "what is a good java book". By now the searching process starts. The query is attacking the first index which is always the index of the internal knowledge database. The query will be stemmed with the same analyzer as the database index. By following that procedure it can be guaranteed, that the query and the index can be compared in a meaningful way. If some suitable answers were found, the program saves them to be shown in the result page. After that, the raw query is attacking the file system indexes, where again the query will be stemmed separately and compared with every file index which was created. All found answers are being saved again. Finally the program looks up if any plugins have been installed to include other resources for the searching procedure. In our case, the Google search engine has been implemented as a plugin using the web services interface. As Google has it's own stemmer and analyzer the raw query is sent to the search engine. Responses are being saved again. By the next step all saved answers are being sent to the user interface which normally is a web browser who visually displays the responses. Now the user has the opportunity to choose an answer which is correct, according to his opinion. By doing so the question and it's corresponding answer are being saved into the (knowledge-) database. Immediately after inserting it in the database, the question will be added to the index of the database, to make it again traceable for another questioner.

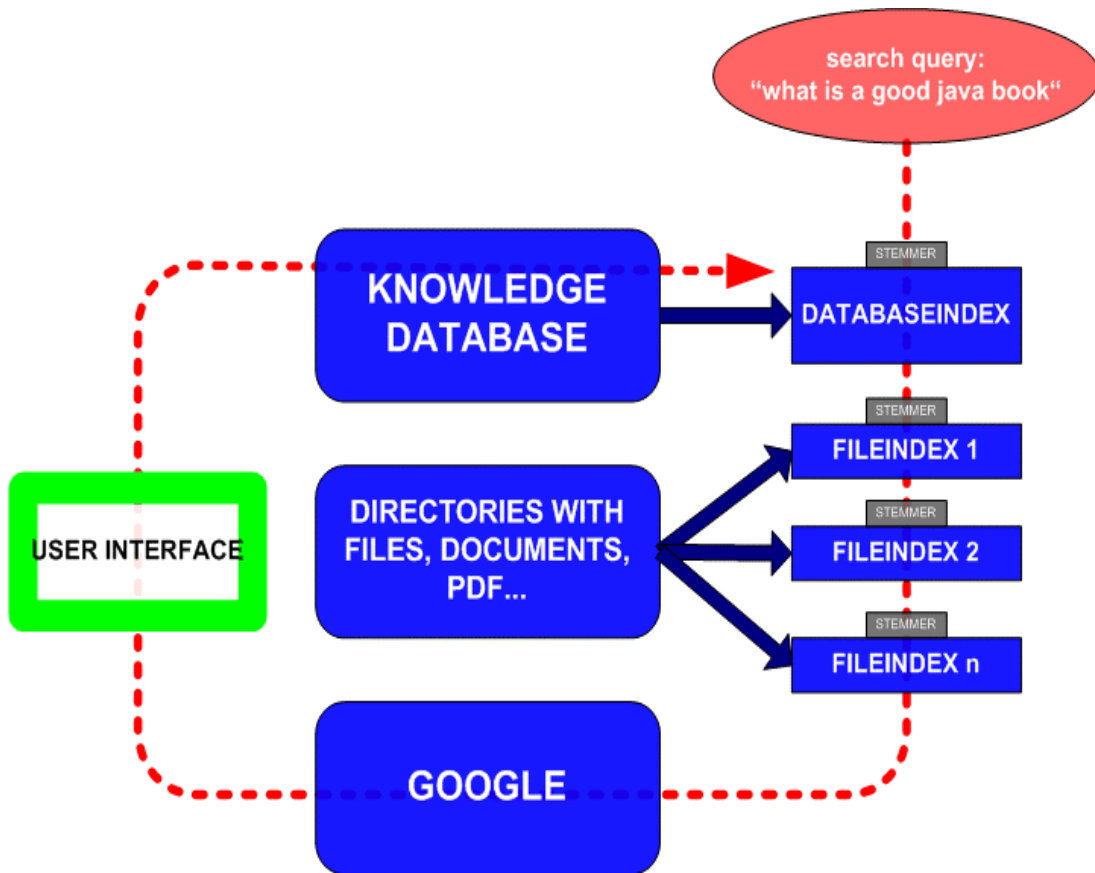


Image 5.15: Searching in Index

5.4 Class Diagram

To following class diagram should help to get an overview of the program structure. The whole code was written in Java [Java] based on the Java Struts Framework [Struts]. Queries to the database are realized via OJB [OJB]. OJB stand for ObjectRelationalBridge and tries to build a bridge between an object model and a relational database model. Details will not be explained here. As one can see in the diagram all classes are derived from the PersistentObject.class. This class holds only information about when and by whom an object was created. The class User is not part of my project, it belongs to the oswp-core-project, but as the prototype is part of the oswp-project it will frequently use this class for identification of a person. The diagram is more or less self describing.

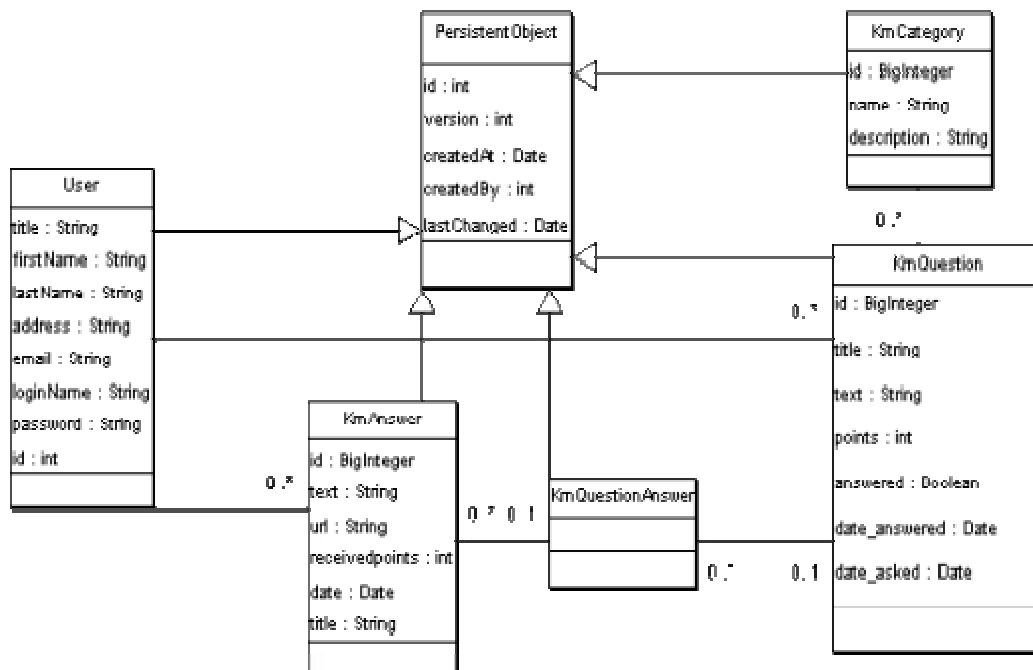


Image 5.16 Class Diagram

- a user can ask several questions.
- a question has one category
- a question has one or more answers
- a user can answer several questions
- an answer has several question (possible but will not be used so far)

5.5 Scenario for a Test

5.5.1 Testing the Usability

The success of this software project will be measured through the eyes of the end users. If the project does not meet the end users' expectations, something has gone wrong. To improve the quality and the usability of the question answering system we need information and facts from a working environment, which we do not have at the moment. We need facts about the comfort using the QAS instead of other alternatives, like searching files manually or searching in Google itself. Users have to accept the new system and they have to integrate it in their daily work. To see if the system reached the consumer acceptance we need some statistics about frequentness of use or about the hit rate of questions. It is possible to receive many statistics automatically by logging the user behaviour and representing it in statistics. A simple implementation of that can be found in the administration area under statistics. This page shows a statistic about every user and its participation in the system. But that is just a train of thought what is really possible. In my opinion, a good testing environment is a mixture of collecting automatically generated statistics and a polling which asks every user about his personal

attitude to the system. It would be no problem to integrate a new web page which appears for instance automatically after one month since the user has been registered. This questionnaire could include the following questions.

- Do you think the system is user-friendly ?
- What do you think about the quality of the answers ?
- How many answers did you ever find which solved your problems ?
- How do you like the design of the platform ?
- Do you use the system as your preferred tool to solve problems ?
- Did you ever find somebody who shared the same problem with you ?
- What is your average impression of the system ?

Of course there are much more ideas. These questions again can have answers from 1 (very good) to 10 (very bad). This makes it easier for the evaluation.

To evaluate a question answering system we choose six factors to judge a system, which are:

- The quality of the answer
- the ability to expand the system
- the ability to enter natural language as a question
- the ability to learn automatically
- the restriction to one domain.

With that method we receive a convenient result if a system meets the requirements of a modern question answering system.

But the more important fact is if the management is satisfied with the introduction of the system. Do they have the feeling that work processes have improved? The management is always concerned about the return of investment. Are the costs of the new system really worth its money? For that reason managers and decision-makers have to talk with their employees about the knowledge management system and finally everything ends up in the statistics managers receive out of the system. Hence, the most important question in the test is, if decision-makers think that the system improved the productivity.

5.5.2 Testing the Code

There are two common test methods to discover errors in the source code itself. There are Black-box and white-box test design methods. Black-box test design treats the system as a "black-box", so it doesn't explicitly use knowledge of the internal structure. Black-box test design is usually described as focusing on testing functional requirements. White-box test design allows one to peek inside the "box", and it focuses specifically on using internal knowledge of the software to guide the selection of test data. Synonyms for white-box include: structural, glass-box and clear-box. While black-box and white-box are terms that are still in popular use, many people prefer the terms "behavioral" and "structural". Behavioural test design is slightly different from black-box test design because the use of internal knowledge isn't strictly forbidden, but it's still discouraged. In practice, it hasn't proven useful to use a single test design method. One has to use a mixture of different methods so that they aren't hindered by the limitations of a particular

one. Some call this "grey-box" or "translucent-box" test design. It is important to understand that these methods are used during the test design phase, and their influence is hard to see in the tests once they are implemented.

To test the code of the prototype black-box testing might be sufficient. White box testing in this case is too time intensive and can hardly be done manually. The aim is to reach every Action Class of the code once. This can easily be reached by clicking on every action-link and by asking various questions and answering it frequently.

5.6 The scientific discoveries and benefits of that paper

One of the mayor scientific discoveries is the integrative effect. This means, that a user only needs one interface, in our case a web browser to access all his resources being used. The prototype has the capability to integrate different sources for the gain of new knowledge. The knowledge existing in many different resources which all have different formats can now be accessed and stay in their natural form. Databases can stay the same way they are, the only thing one has to do is to build a plugin which attacks the database to extract the desired information. A problem which usually is managed by many middleware systems, which build a bridge between existing systems and new implementations. As mentioned before, it is important to have the information and the knowledge in the right format so that it can be processed in a way which brings an advantage and enrichment. In case of the prototype the right form is being realized in a structured knowledge-database.

Another issue is the participation in a playful manner using "points" which can be seen as a substitute for money. By "trading" with knowledge, the offerer of the good receives a virtual counter-value. Knowledge gets a new point of view, as soon as one realises, that knowledge becomes a good which can be traded easily.

Another point which hopefully came out clearly in this paper is, that companies have to think about their knowledge management strategy. Knowledge becomes besides other resources a more important good. By now it has always been in the background, but it is time to come into the front, and to make knowledge management one of the mayor tasks of every CIO or even CEO.

But of course the exclusive scientific discoveries are embedded in the prototype. By using the question answering system in operation the real benefits will show up.

5.7 10 Theses and Future Outlook

- *knowledge management receives not enough attention*
although 60% of all businesses are services knowledge still does not receive enough attention. It is still seen as something intangible. This work should demonstrate that it does not have to be that way
- *knowledge owners have the ledge towards their rival firms*
one reason why consulting companies are successful is, because they have a well operating knowledge system.
- *Question answering will soon become reality*
this work has shown that some question answering systems return usable results. It won't take long until real-time systems even with voice recognition can be found in our everyday life.
- *The problem of knowledge is not that it does not exist, the problem is the accessibility*
The problem today is the half-life period of information. Data that has been saved five years ago on a floppy disk is not accessible anymore. The fast changing technologies without solid standards makes it almost impossible to access all kind of information
- *Common knowledge is too complex to map it in a conventional database*
as the Cyc project has shown, it is too difficult to map our common knowledge to a computer system. The fast successes at the beginning were followed by a not conquerable border to the perfectibility
- *The development environment and framework JavaStruts will become a standard*
the framework which has been used to implement the knowledge management system JavaStruts is an Open Source project from the Apache Group. The struts forum is one of the most discussed groups on the Apache sites. The framework supports the MVC Framework which separates the business logic and the view.
- *Lucene builds a stable reasonable API which can be used for indexing and searching data*
Lucene is also a project from the Apache Group and open source as well. It has an API programmed in Java which can be used in own projects. With Lucene it is possible to index documents fast and easily. The option to use that API was very timesaving, when considering that programming a stemming and indexing software would cost a lot of time to receive comparable results.
- *The internet is a huge collection of unstructured knowledge*
Some companies try to structure the masses of information found in the internet. For instance Google or AnswerBus, which are mentioned in the paper try to discover answers or solutions to questions or problems. Other approaches like RDF are trying to solve the disorder in the internet. Finally we will have a mixture of both. Little order and a lot of confusion
- *Before realizing a knowledge management system with different sources one should carefully think about a common format of all sources*
one of the most difficult tasks to solve was, to bring all kind of different formats of information together to one format. Web-links, different files like PDF or TXT

and discussions in forums all had to come together to “find” space in the knowledge database. A word document, for instance, has another structure than a HTML-file. Without a common format it is just data and not knowledge.

- *Knowledge management systems can only be successful if users are being motivated to participate*

The reason why many knowledge management systems did not succeed was, because users did not see an advantage for themselves when they wrote down their knowledge. Good approaches of software failed because of the little motivation. New developments have to take into account the user participation. A good knowledge management system starts at the individual.

5.8 Facts about the Prototype

The realized Java Code of the prototype has 6710 lines which, if you transfer it to a word document with font size 12, is 140 pages.

To program the whole question answering system I was occupied more than 4 month.

Acknowledgements

I want to thank especially
Alexander Schatten
for the excellent management of the whole OSWP Group.

I also want to thank
Marian Schedenig
and
Gerhard Hipfinger
for the support during the implementation process and of course
my parents
for my humble existence.

References

- [Abdecker 1999] A. Abecker, A. Bernardi, and M. Sintek. Developing a knowledge management technology: An encompassing view on know more. In Proceedings of the 8th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, pages 216–222, 1999.
- [Alavi 1999] Alavi, M. and Leidner, D., (1999), “Knowledge Management Systems: Issues, Challenges, and Benefits”, Communications of the AIS, (1)7 Ames, P. (2000) Personal communication. April
- [Albrecht 1993] F.: Strategisches Management der Unternehmensressource Wissen. Inhaltliche Ansatzpunkte und Überlegungen zu einem Konzeptuellen Gestaltungsrahmen. Frankfurt am Main, Dissertation, 1993
- [Answerbus] Web-based open domain question answering system based www.answerbus.com (last visited 1.12.2003)
- [Babelfish] <http://babelfish.altavista.com> , free web tool for translating languages (last visited 1.12.2003)
- [Baseball 1961] Green, B. et al. ‘BASEBALL: an automatic question answerer’, Proceedings of the Western Joint Computer Conference, 19, 1961, 219-224.
- [Borghoff 1997] BORGHOFF, Uwe M. ; PARESCHI, Remo: Information Technology for Knowledge Management. In SPRINGER: Journal of Universal Computer Science vol. 3, 1997, pages 835-842
- [Bray 1996] Bray, T., “Measuring the Web” Available at http://www5conf.inria.fr/fich_html/papers/P9/Overview.html May 6-10, 1996, Paris, France (last visited 15.12.2003)
- [converter] <http://www.jguru.com/faq/view.jsp?EID=1074237> for Adobe PDF
<http://www.jguru.com/faq/view.jsp?EID=1074234> for Word Documents (last visited 4.1.2004)
- [Copestake 1990] Copestake and Sparck Jones, 1990, Natural language interfaces to databases. The knowledge engineering review. 5(4):225-249

- [cyc] the world's largest and most complete general knowledge base and commonsense reasoning engine: www.Cyc.org (last visited 20.11.2003)
- [Drucker 1996] P.F.:Umbruch im Management. Was kommt nach dem Reengineering Düsseldorf: ECON, 1996.
- [Goldsmith 2000] Goldsmith, J.: Unsupervised Learning of the Morphology of a Natural Language. In Computational Linguistics, 27(2), pp. 153-198, MIT Press. URL: <http://humanities.uchicago.edu/faculty/goldsmith/Linguistica2000/> (last visited 3.10.2003)
- [Götz 2000] K.: Managementkonzepte. Wissensmanagement. Zwischen Wissen und Nichtwissen. München Mering:Rainer Hampp Verlag, 2000
- [Grant 1996] R.M. Toward a knowledge based Theory of the firm. In: Strategic Management Journal, 17 (winter special Issue 1996), S.109-122
- [Hirschman 1998] L.Hirschmann, Natural Language Question Answering: The View from Here, Natural Language Engineering 1, 1998 Cambridge University Press
- [Hoffmann 1989] Einführung in die allgemeine Managementlehre. Interdisziplinäre Abteilung für Wirtschafts- und Verwaltungsführung, 5.Auflage Wien: Fachverlag an der Wirtschaftsuniversität Wien, 1989
- [Java] Java technology is a portfolio of products that are based on the power of networks and the idea that the same software should run on many different kinds of systems and devices. <http://java.sun.com/> (last visited 4.3.2004)
- [Knögler 1999] KNÖGLER, Bernhard: Wissensauffindung in verteilten Systemen, Institut für Informationsverarbeitung und Computergestützte Neue Medien, Technische Universität Graz, Diplomarbeit, 1999
- [Krallman 2000] Hermann Krallman (Hrsg.), Wettbewerbsvorteile durch Wissensmanagement, Methodik und Anwendungen des Knowledge management, 2000 Schäfer-Poeschl Verlag
- [Lehnert 1997] W.Lehnert 1977 A conceptual theory of question answering. In proceedings of the fifth international Joint Conference on Artificial Intelligence, pages 158-164.
- [lucene] An Apache Jakarta Projekt Lucene: <http://jakarta.apache.org/lucene> (last visited 3.2.2004)

- [Maas 1996] Maas, D.: MPRO . Ein System zur Analyse und Synthese deutscher Wörter. in R. Hauser (ed.): Linguistische Verifikation, Max Niemeyer Verlag, Tübingen, 1996.
- [Mayfield 1999] Mayfield, J., McNamee, P. and Piatko, C.: The JHU/APL HAIRCUT System at TREC-8. In Proceedings of the Eighth Text Retrieval Conference (TREC - 8), NIST Special Publication 500-246, pp. 445-451.
- [Moulinier 2000] Moulinier, I., McCulloh, J. A., Lund, E.: West Group at CLEF 2000: Non-English Monolingual Retrieval. In Peters C. (Ed.): Cross-Language Information Retrieval and Evaluation, Workshop of the Cross-Language Evaluation Forum, CLEF 2000, pp. 253-260, 2001.
- [opencyc] the world's largest and most complete general knowledge base and commonsense reasoning engine: the OpenSource Project of Cyc www.OpenCyc.org, (last visited 20.11.2003)
- [OJB] ObjectRelationalBridge (OBJ) is an Object/Relational mapping tool that allows transparent persistence for Java Objects against relational databases. <http://db.apache.org/ojb/> (last visited 3.3.2004)
- [Piotrowsky2000] Piotrowsky, M. NLP-Supported Full Text Retrieval. Master's Thesis, University of Erlangen april 2000, <http://www.linguistik.uni-erlangen.de/~mxp/Magister/ma-as-report.pdf>
- [Porter 1997] Porter, M. F.: An Algorithm for Suffix Stripping. In Program, 14(3), pages 130-137, 1980. Reprint in: Sparck Jones, K. and Willett, P. (Eds.): Readings in Information Retrieval, pp. 313-316. Morgan Kaufmann Publishers, San Francisco, CA, USA. 1997.
- [PorterStemmer] English Stemming Algorithm by Porter, M.F. <http://www.tartarus.org/~martin/PorterStemmer/def.txt> (last visited 11.12.2003)
- [Probst 1997] G.J.B. /Deussen, A.: Wissensziele als neue Managementinstrumente, Wiesbaden In: Gabler's Magazin (08.1997), S6-9.

- [Rijsbergen 1979] C.J. Rijsbergen, Information Retrieval, <http://www.dcs.gla.ac.uk/Keith/Preface.html> (last visited 3.3.2004)
- [Savoy 2002] Savoy, J.: Cross-Language Information Retrieval: Experiments Based on CLEF 2000 Corpora. Information Processing & Management, to appear, 2002.
- [Schatten 2003] Alexander Schatten, Closing the Gap: From Nescience- to Knowledge Management, Institute of Software Technology and Interactive Systems, February 2003
- [Schatten2 2003] Alexander Schatten, Franz Inselkammer, A Min Tjoa, System Integration and Unified Information Access using Question Based Knowledge Management Strategies, Institute for Software Technology and Interactive Systems Vienna University of Technology, June 2003
- [Simmons 1965] Robert F. Simmons. *Answering english questions by computer: a survey*. Communications of the ACM, 8(1):53--70, January 1965.
- [Struts] Java Struts Framework, The goal of this project is to provide an open source framework for building Java web applications. <http://jakarta.apache.org/struts/> (last visited 3.3.2004).
- [Spiegler 2000] Israel Spiegler, Knowledge Management a new idea or a recycled concept, Volume 3 Article 14 June 2000, Communications of the Association for Information Systems
- [Wechsler 1997] Wechsler, M., Sheridan, P., and Schäuble, P.: Multi-language text indexing for internet retrieval. In Proceedings of the 5th RIAO Conference, Computer- Assisted Information Searching on the Internet, Montreal, Canada, pp. 217-- 232, 1997.
- [Wiig 1999] On Conceptual Learning, Elisabeth H. Wiig & Karl M. Wiig, Knowledge Research Institute, Inc. Working Paper 1999-1
- [Willke 1997] H.: Wissensarbeit. Organisationsentwicklung. Stuttgart: Lucius & Lucius, 1997