SYSTEM INTEGRATION AND UNIFIED INFORMATION ACCESS USING QUESTION BASED KNOWLEDGE MANAGEMENT STRATEGIES

Alexander Schatten, Franz Inselkammer, A Min Tjoa
Institute for Software Technology and Interactive Systems
Vienna University of Technology
{schatten, inselkammer, tjoa}@ifs.tuwien.ac.at

Abstract

A question based knowledge management system with the capability to integrate heterogeneous sources of information and knowledge and which nonetheless acts like a single coherent system with only one user interface is introduced. Especially this interface and the easy access to different information-resources makes it comfortable for users with even low IT knowledge to find their way through complex and scattered information landscapes. This paper will particularly describe the integrative effects and the aspects of creating the knowledge base: Users pose questions, answers are created by the system resorting to the internal knowledge base, external information systems and eventually also involving the systems user-base. This mechanism is assisted by a meaningful scoring system. Finally the external resource interface is being explained as well as the time and cost saving factors.

1. Introduction

Lack of information and data is not the main problem in contemporary business and university environments. The opposite problem arises as Alavi et.al. [2] describe:

> *"The concerns related to information were primarily associated with a desire to avoid overloading already taxed users with yet more information. The concern was as much about the new information that would now be available as it was about eliminating 'old/wrong data' or knowledge that was no longer valid. This supports Courtney et al's [3] assertion that 'omitting the unimportant may be as important as concentrating on the important' in determining what knowledge to include in KMS."*

The information overload problem to evaluate the quality of information sources has different reasons: First of all, information landscapes are usually not planned hierarchically, but are grown by needs. This leads to a *"hodgepodge of executive IS, group-support systems, intranets, decision-support systems, and knowledge-based systems"* as Stewart et.al. [9] express it. These scattered information sources are hard to consolidate, as for different reasons they are often implemented by different departments and different persons. Alternatively a thorough bureaucratic approach is not advantageous, as users tend to avoid collecting information, if it is too difficult to integrate it in existing systems (which was the problem of the "mainframe age" in the seventies and eighties). An unnecessary loss of information and knowledge might be the consequence then.

So this "hodgepodge" of information system boils down to two essential problem fields: The first one involves the managerial and quality assurance issues as well as, technical issues of the systems involved, whereas the second problem concerns the accessibility of those resources by the user community. We should put in parantheses that information may be very useful also for users outside the original targeted community! This is particularly problematic, when rather unskilled co-workers have to use a broad range of information systems to solve their daily problems. Different IT strategies have been suggested to get a grip on the various aspects of information and knowledge management (KM) issues. But KM approaches tend to fail by various reasons, among others:

?? Lack of Participation
?? Lack of a quality criterion for knowledge stored in the system(e.g. which old data can be removed)
?? Uasability for unskilled employees
?? System design decisions (e.g. Is the system designed for a closed user group, or for "all" possibly interested users?, Is the system proactive or reactive in knowledge acquisition?)

In our opinion in *proactive knowledge acquisition* it is important to answers questions, such as: who knows what knowledge will be really needed in the daily work? Who motivates *all* users to provide their knowledge? How is this knowledge evaluated, and so on. Schatten et.al. [7] place their emphasis in the importance of *nescience* in any knowledge and information driven business:

> *"The term knowledge management obviously suggests the necessity to deal with information and knowledge, but […] direct access to knowledge is difficult by many reasons. […] when knowledge is generated and applied (!) society proceeds one step higher in system complexity; the nescience, the insecurity increases, new problems arise, and generally spoken the system risk grows".*

The consequence is, that the management of nescience, hence the management of system risk is one of the driving factors for successful problem solving, and should be considered when building KM environments. A *question based system* is suggested in Schatten et.al. [7], and details about the KM aspects can be found there. However, there is a second essential aspect in KM as discussed earlier, namely the *usability and integration of information and data sources*. A second important advantage of the question based reactive approach is, that in such a system, *information integration* and *unified access* also for *unskilled employees* can be an integrative aspect. Consequently in the following sections these aspects are illustrated in more detail.

2. 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 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 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. (For instance a user could have a favourites list of the information or questions lately or frequently visited, as soon as she is logged in.)

3. Description of the User Interaction Scenario

3.1. Introduction

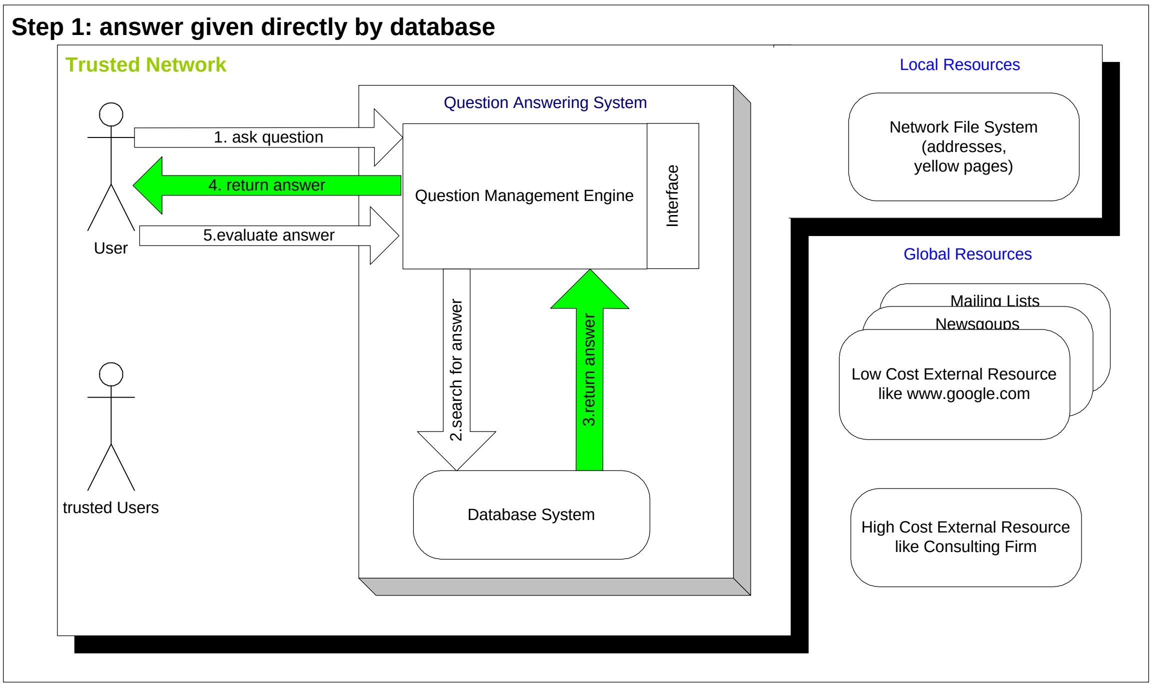
In this section, the concrete scenario of a user query will be described in detail as a combination of the inclusion process of different internal and external information sources and on the other hand the building of a knowledge repository following the users interaction with the system.
The users view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

3.2. 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 the hit ratio for an answer to a question is expected. 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 [4]).

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.).

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.

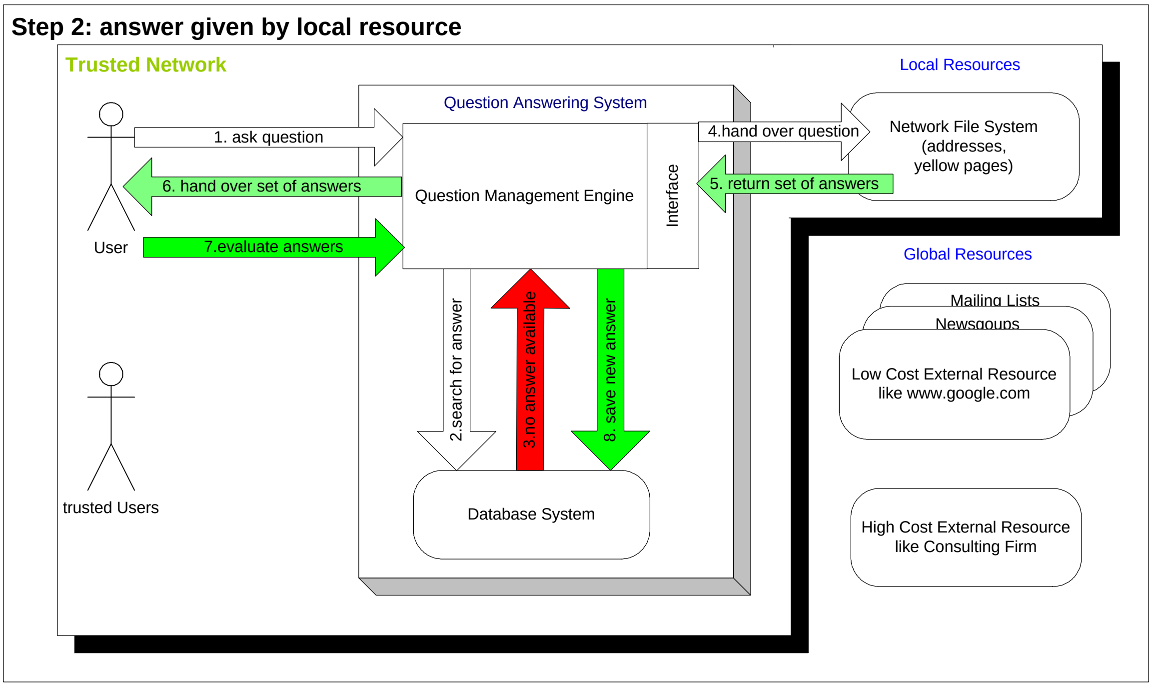


3.3. Step 2: Answer given by Local Information Resource

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.).

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 planed 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.

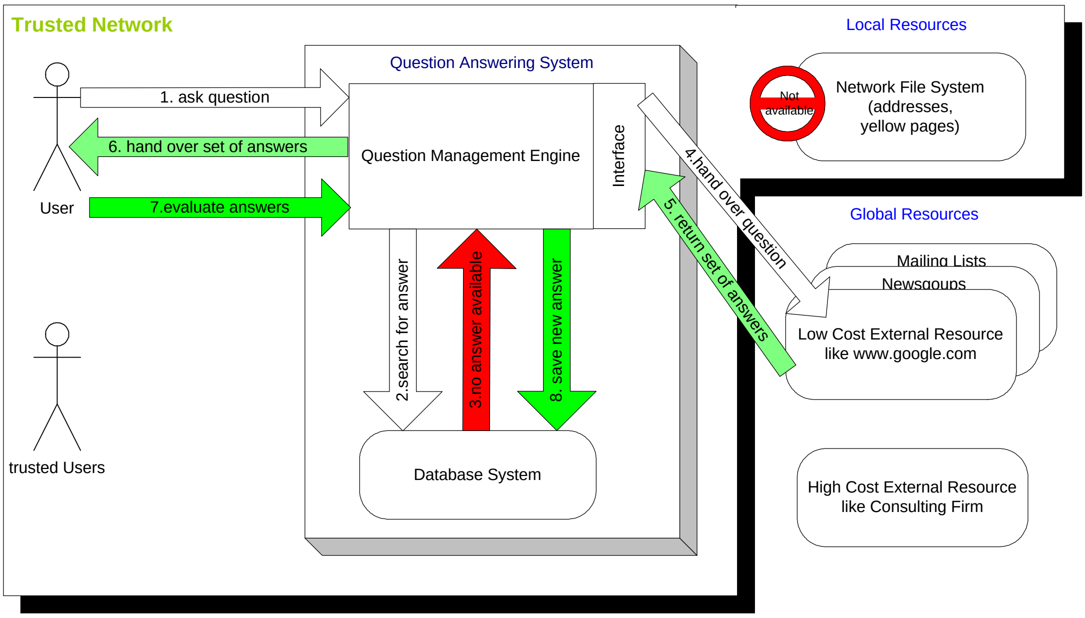


3.4. Step 3: Answer given by a Global (External) Resource

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. Usually the link should sufficient at the moment. 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.



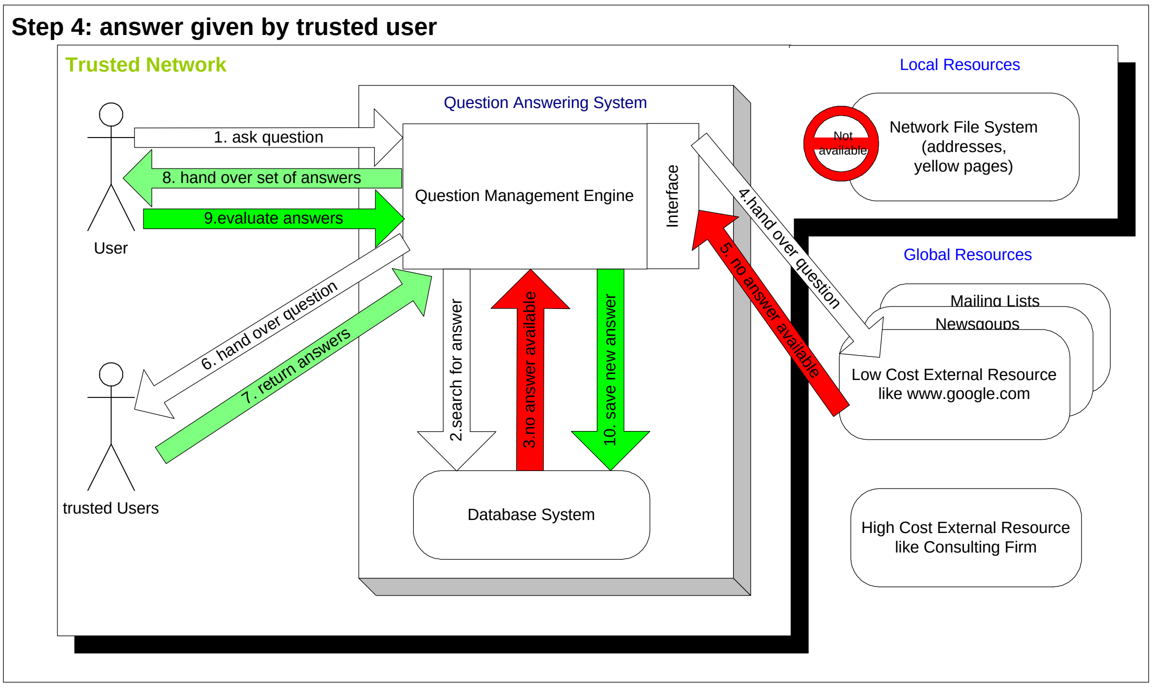
Step 3: answer given by global resource

3.5. 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 be answerable by the use of a global resource (4. and 5.). Obviously (as described in step two) the answer can not 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 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 her 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.

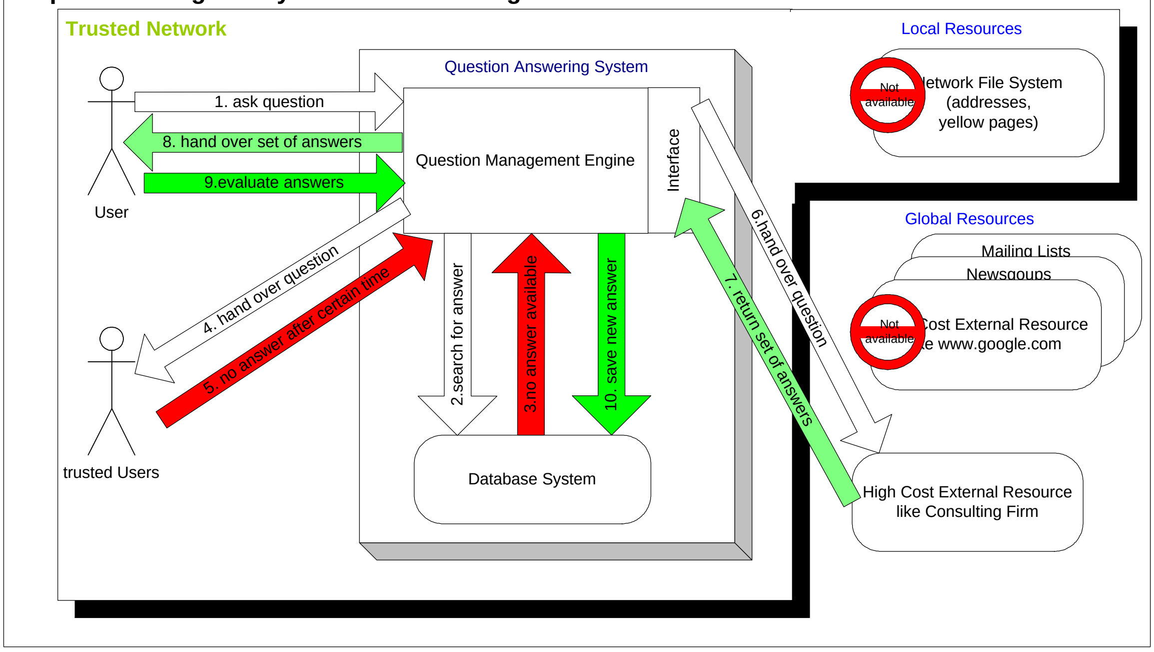


3.6. Step 5: Management Activities

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.



Step 5: answer given by external consulting firm

3.7. 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. 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.

4. 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[1]. 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 one could be the interface to the global resources, etc. 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. 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." (Rus et .a [6])*

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. A specific motivation has to be present to share knowledge. 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. 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.

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,

---

[1] 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.

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. 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. 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. 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 [1]:

> *"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 et.al. [1]*

6. Outlook and Future Works

A prototype of the suggested KM question and answering system is designed and will be implemented as part of the redesign of the Open Science Workplace. This is a joint effort of the Vienna University of Technology and the University of Kerman to provide an open source project communication, management and monitoring platform. This Q+A system fits naturally into such a CSCW environment as it supports problem solving and tracing. Details about the Open Science Workplace Project which is funded by the Austrian Federal Ministry of Education, Science and Culture can be found at http://www.oswp.info and in Schatten et. al. [8].

7. References

[1] 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.

[2] Maryam Alavi and Dorothy E. Leidner. Knowledge management systems: issues, challenges, and benefits.Communications of the AIS, 1(2es):1, 1999.

[3] Courtney, J., D Croasdell, and D. Paradice (1997) "Lockean Inquiring Organizations: Guiding Principles and Design Guidelines for Learning Organizations," Proceedings of the AIS '97 America's Conference on Information Systems

[4] Martin Braschler and Bärbel Ripplinger. Stemming and decompounding for german text retrieval. In Proceedings of the ECIR, pages 177–192. Springer, 2003.

[5] Manfred Füllsack. Leben ohne zu arbeiten? Zur Sozialtheorie des Grundeinkommens. Avinus Verlag, Berlin, 2002.

[6] Ioana Rus and Mikael Lindvall. Knowledge management in software engineering.IEEE Software, pages 26–38, May/June 2002.

[7] Alexander Schatten, Stefan Biffl, and A Min Tjoa. Closing the gap, from nescience to knowledge management. In Proceedings of the Euromicro Conference. IEEE, 2003.

[8] Alexander Schatten, A Min Tjoa, Amin Andjomshoa, and Hassan Shafazand., Building a web-based open source tool to enhance project management, monitoring and collaboration in scientific projects. In IIWAS Conference, 2001.

[9] Kathy A. Stewart, Richard Baskerville, Veda C. Storey, James A. Senn, Arjan Raven, and Cherie Long. Confronting the assumptions underlying the management of knowledge: an agenda for understanding and investigating knowledge management. ACM SIGMIS Database, 31(4):41–53, 2000.