The “Spree” Expert Finding System

Florian Metze, Christian Bauckhage and Tansu Alpcan
Deutsche Telekom Laboratories
Berlin, Germany
{florian.metze|christian.bauckhage|tansu.alpcan}@telekom.de

Kathrin Dobbrott
Fachhochschule für Technik und Wirtschaft Berlin
Berlin, Germany
kathrin.dobbrott@gmx.de

Caroline Clemens
Zentrum Mensch-Maschine-Systeme
Technische Universität Berlin
caroline.clemens@zmms.tu-berlin.de

Abstract

This paper proposes a system to facilitate exchange of information by automatically finding experts, competent in answering a given question. Our objective is to provide an online tool, which enables individuals within a potentially large organization to search for experts in a certain area, which may not be represented in company organization or reporting lines. The advantage of the proposed system over standard forums or group-ware systems is that full-formatted questions can be compared to stored qualification profiles, which were automatically derived from documents, without Human search effort, and possibly refined manually. This allows us to find competent colleagues (or helpful literature such as How-Tos) for a given problem in a single step, and without intermediate iterations. The system is symmetric in that it does not distinguish between “questioners” (asking questions) and “experts” (answering them), therefore forming a “community” of users, which are distributed over an ontology covering the total knowledge. This ontology can either be given (i.e. in the form of an organizational chart), or it can be derived from the experts' knowledge.

1. Introduction

Our expert finding tool is a web-browser based application, mainly targeted at larger corporations, in which discovery and dissemination of available information is often a significant problem: users (high-level managers, employees in a different division, ...) need to be able to find competent experts easily, and without too many steps. Currently this is usually done by asking a few well-connected colleagues, or “networkers” for advice, who might pass on the request to other networkers, resulting in the creation of a community, which is vital for the survival of the organization, but also results in frequent disruptions for many workers to handle colleagues' requests such as “who knows Y”.

In this paper, we use the term “expert finding” in the same sense as NIST's Text REtrieval Conference (TREC) series (http://trec.nist.gov), as our goal is to identify an expert using documents describing his qualification profile.

spree tries to improve knowledge management by introducing communities and “social software” [15] in the process, which has recently received much attention as a social model for information dissemination on the World Wide Web. The term “social software” means web-based solutions, which facilitate the creation of social networks and foster exchange of information therein. They are based on the principles of self organization, absence of control, and volunteered participation. Social feedback in the form of social ratings (number of links, comments, etc.) plays a significant role in these networks, as members do not only consume information, but also actively provide and organize information. As spree connects questioners with a competent expert, it is as easy as possible for community members to provide or retrieve information.

spree is not meant for answering everyday queries for factual information such as “what companies were acquired by IBM”, instead we envision it to be used for queries requiring a deeper understanding and therefore profiting from discussion with an expert, such as an assessment of why IBM bought these companies. Need for this type of information arises frequently in larger organizations and it can not usually be satisfied by simply pointing the user to a document, but a first understanding can be reached easily by talking to someone who has dealt with IBM's strategy before. spree's target therefore is to find this person, or at least someone close to him.

During registration, every member of the spree community is assigned a competence for every node of an ontology tree spawning spree's domain of knowledge. These competence scores can now be used to identify the best expert(s) to answer a question falling into specific categories. The basic process therefore is as follows:

A spree community member (in its "questioner" role) types a question into an entry field on a web page, which looks as if he was posting to a support forum. spree transforms the question into a "bag of words" representation, determines the closest node(s) of the ontology, and contacts the best experts, i.e. the five users having most knowledge about the query. If one accepts, spree will establish an online chat connection between questioner and expert to discuss the problem. While the "questioner" is waiting for an expert to join the chat, or if no suitable experts are available (e.g. because they are offline), spree will show the most relevant entries from an internal database, which in the form of frequently asked questions (FAQs) might provide the answer without actually disturbing an expert. The strategy to follow during the matching step can be adjusted as needed, so that immediately available (but less competent) experts are preferred over more competent experts, which may be out of office and not available for the next few hours, or vice versa. Queries can also be re-submitted with a manually provided classification to reach different experts. At the end of a chat or after reading FAQs, the expert and documents can be rated by the questioner to improve the quality of the matching and to provide incentives to the members of the community.

Figure 1 shows the overall process: a community member's request is forwarded to a relatively low number of well-qualified experts, one of which will accept the query and join the member in a chat. spree is currently being implemented as a prototype application at Deutsche Telekom Laboratories in cooperation with DAI Labor at Technische Universität Berlin.

This paper presents a description of the spree concepts, it discusses the current implementation in the form of a web application, and presents results of a requirements' analysis and user studies conducted during the implementation of the prototype software.

2. Knowledge Management

For most organizations today, their most valuable assets are not property or buildings, but employees and their skills. Employees are also an important knowledge sources in enterprises [5]. Knowledge management tools help to find, store and distribute knowledge. There are two approaches of knowledge management systems: the *codification* approach focuses on explicit knowledge that can be stored and the *personalization* approach, which puts priorities on implicit knowledge and the interaction between users [8]. spree follows the personalization approach.

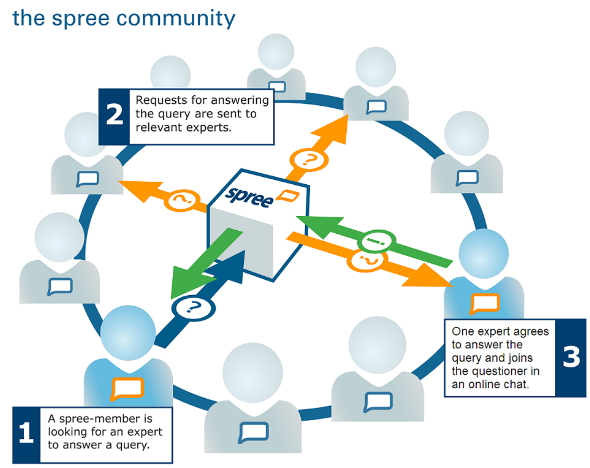

Figure 1. Finding experts with spree**.**

2.1. Knowledge Management in Organizations

Knowledge exchange requires a network between users as a structure for the collaboration. There is no hierarchy in such a structure so that all members are connected on the same level. In that way, an additional network is formed across internal structures of organizations or locations. The network is an enrichment, not a displacement of existing structures [9].

A successful knowledge management system therefore has to accomplish the following [3]:

1. Offering and optimizing the access to available knowledge at the right time and the right place.
2. Storing and archiving available knowledge so that it stays in the company's memory even if the original expert has left.
3. Optimizing the use of knowledge by deleting unused knowledge and integrating new useful knowledge.
4. Generating new knowledge and enhance competences.

Information technology can help with storing, searching, and presenting knowledge. Additionally, the connection of knowledge to its context (e.g. social structures or relationships) has to be considered.

2.2. Current expert finding tools

The World Wide Web offers many possibilities to get questions answered by both document search and other Humans. This section discusses other existing systems. Their approaches to presenting questions and answers can be categorized as follows:

Bulletin and question boards or "forums" are web applications for holding messages such as questions, answers, discussions and posting user generated content or comments. Messages are displayed either in chronological order or as threaded discussions. They can be differentiated into two types: get an answer for a problem or find the right person to get a good solution.

The first type is represented in web sites such as http://qna.live.com, http://iq.lycos.de, http://wiki.answers.com, http://www.wondir.com, http://www.gutefrage.de, http://uclue.com, or http://askville.com. Questions will usually be classified according to keywords or categories provided by the user. Communication in forums works asynchronously, i.e. some time gets lost until an answer is received. Every person in a forum can answer, which means that in many cases unqualified users give well-meaning, but incorrect, answers. Some of these question boards have special features like an installable toolbar for browsers (http://iq.lycos.de), automatic classification of questions (http://www.wondir.com), a presentation of similar questions and extern links, or sending unanswered questions as email to experts which could have the knowledge to answer the question (http://www.gutefrage.de).

The second type is given by e.g. http://allexperts.com and provides set categories where the questioner can choose and find the right person for his problem and ask the expert directly via email. The registered experts have been admitted and categorized through an application system.

Search engines with human guides: *ChaCha* (http://www.chacha.com) is similar to ordinary search on the Web using e.g. Google, but involves more users in the search. If a questioner does not find any good results on the Web, he or she can ask a "guide" in order to get better results. Thousands of search experts (guides) therefore have to be available (and often paid), yet results depend on the web-browsing skills of the experts, not on any real insight into the original problem.

Stand-alone applications: *Illumio* (http://www.illumio.com) is a group question and answer tool incorporating statistical matching. *Illumio* finds the best group member to answer a question by automatically routing the request to those who have direct interest or expertise in the topic at hand. *Illumio* distinguishes between private and public groups and therefore it is suitable for project or team work in enterprises. Users immediately know when they get a request as it will be shown in a notification box. *Illumio* however requires users to install dedicated client software in order to use the system.

Messengers with discussion platform like Skype, ICQ or MSN integrate the possibility to create own discussion platform as chat rooms, where interested parties can discuss about a specific topic. These chat rooms are classified using categories or tags. Experts with the right knowledge about a specific topic however can only be found by chance in this approach.

Network and contact management portals: to get a solution for a problem the user can look for an expert on several commercial networking platforms on the Web like http://www.bizwiz.com or http://www.xing.com. These are an online listing of people, their competences and contact information. Users have to manually register and describe their skills themselves. These platforms are developed to connect users with business and commercial interests. Users will tend to overrate their skills and expertise to get into contact with more potential clients [7].

Email analysis: *Xpertfinder* [4] is a tool that monitors email traffic as well as several kinds of files like PDF or Word documents located on servers within the enterprise. It analyzes authorship, content and communication structures from these sources and derives a mapping from persons to predefined expertise fields. The main focus of *Xpertfinder* is analyzing email exchange to create personal profiles of experts, but can also be used to find experts [7].

2.3. Knowledge Management using spree

spree's approach tries to solve the four knowledge management problem areas identified in Section 2.1:

1. The main functionality of the spree system is to facilitate access to existing knowledge. This is achieved through means of an expert. Therefore, a large population of experts has to be actively participating in the spree community.

2. spree tries to facilitate the conservation and archival of existing knowledge by making it easy for experts to create offline "notebook entries" from online chats they had with questioners. Our user experience research (see Section 5) showed, that experts do not want to be bothered with "frequently asked questions" (FAQs), so spree tries to answer these questions by

showing highly rated notebook entries before connecting to an expert.

3. Users can edit their notebook entries in order to make them more readable or to improve matching. Questioners can rank experts and notebook entries according to their perceived usefulness.

4. To facilitate exchange of ideas, generate new ideas and train users, spree uses synchronous ways of communication between users, i.e. chat, whenever possible.

While other web-based approaches to knowledge management or question answering discussed in Section 2.2 such as http://www.wondir.com or http://www.gutefrage.de also automatically categorize questions, they forward questions to all experts who have associated themselves with a given keyword. The matching between experts and questions is thus rather weak. *Illumio* uses groups to further narrow down the selection, but as of now only spree uses a matching algorithm to assign a question to a low number (usually about five) of experts, i.e. all other users. Experts therefore have less load and can spend more time answering questions. While not implemented at the moment, a further differentiation of spree from other expert finding systems will be the inclusion of a ranking system for both experts and notebook entries. We expect that rankings will further help to improve matching and act as an incentive in social networks.

Illumio and *Xpertfinder* automatically generate a knowledge profile for users from given data. *Illumio* relies on Google Desktop Search (GDS) for that purpose. spree can index a given set of documents provided by the user, so users can submit the documents they specifically *want* to be taken into account without the inconvenience and risk of having their profile generated on a large corpus of data, which may be unrelated to their professional qualifications. spree attaches the keywords extracted from a user's documents to an ontology, providing an abstraction and generalization layer in the process. The ontology can be given, e.g. by the organization, or it can be generated from users' documents, therefore allowing for skill management within a company (see Section 4).

spree is unique among current web-based solutions in that it offers online chat between users to immediately answer questions and also offers a discipled approach to finding the best people to chat with, in order to foster exchange of knowledge.

3. System Architecture and Implementation

Recent developments such as widespread availability of broadband Internet access and standardization of Web technologies have lead to an explosion of Internet services and applications which are conveniently labeled as *Web 2.0 technologies* [10]. Consequently, the Web browser has become a platform for applications assisting or even substituting the ones that traditionally run on a personal computer.

The implementation of spree follows the basic Web 2.0 paradigms. The system's functionality is implemented on a central server and we utilize the Web browser as the client application, benefiting from its widespread availability and familiarity. We rely on recent Ajax libraries and toolkits to not only facilitate communication between the spree-client and the spree-server but also to develop a rich user interface for increased usability. Ajax' asynchronous communication model between the server and clients, plays a crucial role in running our rich client application within the Web browser. Furthermore, we use Dojo http://www.dojotoolkit.org to provide a variety of standard GUI elements, e.g. modal windows, tabs, and trees.

The web application server is realized using the open-source TurboGears framework [13]. Additionally, our server-side framework integrates a standard database (MySQL), a web server (Apache), and the spree matching engine (see Section 4).

In addition to sending periodic updates on incoming queries to each client, the spree-server also facilitates real-time communication between users via instant messaging (chat). Since the client-side is fully implemented within the Web browser all client-to-client communications have to pass through the server where the chat logs are stored for future private use of the clients. Due to lack of mature push technologies on the Web, the client-server communications are currently handled via Ajax-based polling mechanisms. Another important feature of the spree system is its scalability. To ensure a high quality of service to the users we are investigating intelligent caching and polling solutions, which decrease the peak loads on the server. Furthermore, standard load balancing solutions such as a reverse proxy are being implemented as part of spree.

4. Matching Algorithm

In this section, we briefly sketch our algorithmic approach to expert finding. It incorporates topic specific ontologies, which allow for mapping queries and expert profiles into abstract semantic spaces. There, matchings can be computed efficiently, and results obtained from experimental evaluations demonstrate the viability of the approach.

4.1. Algorithm Design

Our system is based on the use of an ontology tree where each node corresponds to a (sub-)topic or area of knowledge. Consequently, the fields of expertise (profile) of the participating experts can be thought of as subtrees of this ontology. Since user queries, too, can be mapped to subtrees, assigning queries to relevant experts becomes a graph matching problem. Serializing the ontology tree allows us to consider simple dot products on the resulting vector space in order to efficiently address this problem.

While ontology spaces allow for using well established methods from linear algebra, their dimensions are considerably smaller than those of the commonly used term-by-document spaces. In an offline preparation step, we first associate each node of the tree with a corresponding *bag of words* harvested from related Internet resources. Subsequently, all bags of words are pruned. In the vertical direction, we unify the bag of words of a node with the ones of its children. Starting with the nodes above the leaf level, this is repeated in a bottom-up manner until the root node is reached. In the horizontal direction, we find the overlapping words among all children of a node and subtract them correspondingly. Altogether, this procedure removes ambiguities, emphasizes the hierarchical nature of the underlying semantic structure, and accelerates subsequent processing.

Since expert profiles (provided by means of resumes, webpages, notebook entries, etc.) as well as user queries can be turned into bags of words, too, they can be matched against the nodes of the ontology. Using the TF/IDF measure from information retrieval [14] for comparing bags of words, we apply a top-down procedure to map profiles and queries onto subtrees. Once the registered experts $e_1, \ldots, e_E$ and a query q have been mapped to subtrees, we determine associated vectors $\mathbf{v}_{e_i}$ and $\mathbf{v}_q$ whose entries are set to 1 or 0 depending on whether or not the corresponding tree node is related to the expert's knowledge or the query, respectively. Matching scores then result form the weighted dot product

$$m(q, e_i) = \mathbf{v}_q^{\mathrm{T}} \mathbf{W} \mathbf{v}_{e_i} \quad (1)$$

where the weight matrix $\mathbf{W}$ allows for incorporating further contextual information [1]. Subsequently, the query will be assigned to the experts with the highest scores.

4.2. Experiments

Encouraged by initial tests with small vocabularies and ontologies of modest size [1, 2], for this paper, we conducted extensive experiments with the database of knowledge collected by the open directory project (http://www.dmoz.org). This ontology was pruned to a tree of 1318 nodes distributed over three levels. Several million topic related keywords were obtained from harvesting HTML documents retrieved from the Yahoo! search engine. 80% of the documents assigned to each node were used for populating the tree while the remaining 20% served as an independent test set. Exemplary queries were created by randomly choosing a topic and a number of keywords from the corresponding bag of words in the test set. Also, we randomly generated different sets of experts, where each expert was assumed knowledgeable in one to five topics. For each query q, we computed the expert peering, i.e. the set of experts with the highest matching scores. The ground truth vectors served to determine the set of experts that should have been returned for that query. In order to assess *how many of the appropriate experts were retrieved* and *how many of the retrieved expert were appropriate*, we averaged *recall* and *precision* over 1000 queries. We considered different numbers of query words and varied the sizes of the sets of experts. For our experiments, the server was installed on a dual 2.2GHz AMD Opteron 2214 workstation and ran a non-optimized Python implementation of our matching engine.

With these settings, our system yielded a precision of up to 85% and a recall of up to 80%. It was noticeable that the precision consistently exceeded the respective recall for all considered parameterizations. Regarding our application, this is in fact a desirable property.

With respect to average processing times, we found that even for less favorable settings the processing times resulting from our Python implementation do not threaten real time constraints, as they do not exceed network delays under good conditions.

5. Usability Design Process

Assuming high quality expert-finding results, the success of our system crucially depends on the quality of the user interface. We therefore conducted extensive user studies in parallel to the development of the algorithm and the implementation of the backend server, including usability experts early in the design process.

5.1. Aims and Design Principles

The spree system is to be used by a group of peers, e.g. employees within a company, or other institutions. Users are presumed to be computer-literate, comfortable with Internet use and to have good English language skills. The usability design of the system has to ensure that the system is intuitive to use for users of this target group. One main characteristic of the system is the idea of symmetry, i.e. users can and should submit questions *and* answer questions as experts. The interface should convey that idea and encourage the users to be active in both ways. Finally, users need to be available as experts as much as possible. Therefore, the system must entice them to be logged in to the system, so that they can be reached while they carry on with their everyday work. The system for now is available in English only. To ensure consistent wording, text to be displayed on screen is checked by usability experts.

The users' involvement is paramount to the success of the spree system. Social dynamics will then form a community that develops its own culture. Interaction design has to lay the foundations for a respectful and comfortable etiquette, our studies showed that handling of identity (anonymity) and the function for rating experts are the most sensitive issues with this respect. The design of the spree interface conforms to the following Web 2.0 principles [10]:

1. The web as a platform: the spree system acts as an intelligent broker, fostering an ethic of cooperation.

2. Harnessing collective intelligence: users are encouraged to document their knowledge in notebook entries, indexed by the system to identify relevant experts.

3. Extensibility and modular design: while our prototype interface is tailored towards PC-based interaction, our architecture also supports inclusion of (VoIP) phones, mobile phones, text messaging, etc.

4. Rich user experiences: our implementation uses Ajax technologies in order to deliver a full scale interactive application that can be run in modern web browsers.

5.2. Functionality and Implementation

The main functions of the current prototype are:

Personal start page: allows to submit, answer or decline questions delegated to the user. Users will shortly be able to choose among three different interface designs providing the same functionality, but differing in layout and style of interaction:

1. Web search interface: queries are entered in a standard text box and query assignments are presented as individual tabs; the user has to choose manually between searching in notebook entries and getting connected to an expert (see Figure 2 for a mock-up design).
2. Chat client: incoming and outgoing queries are presented in several tabs of a chat window and queries are entered in an empty chat window; this retains a consistent "chat" affordance at all times.
3. Alternative interface: we will also evaluate the merits of a search interface metaphor inspired by the controls of e.g. a portable MP3 music players, in order to increase the "fun" aspect of the spree community.

Set preferences: choices and settings for alerts about incoming queries or updates (visual, audible, etc.).

Set expert profile by uploading papers, URLs, or resumes for indexing (adding or replacing existing information) or by inspecting and manually editing the competence tree, i.e. the personal ontology subtree.

Answer a query either by typing into a chat window, or by dragging a notebook entry into the chat window.

Rate experts: after direct interaction with an expert, the questioner can rate the session, giving credits to the expert. At the end of a chat, the questioner can also rephrase the question, if the answer was not satisfactory.

Save and edit notebook entries: the user can store the chat to a notebook where (s)he can edit it and make it available as an answer to future searches in order to avoid manually answering them again, generating FAQs in the process.

5.3. Testing and Evaluating

We utilize several usability testing methods in order to investigate if the system fulfills the criteria of effectiveness, efficiency and satisfaction according to [6] and to get indications as to how its performance can be improved.

Following [12], we started preliminary, iterative user tests, allowing us to correct simple usability problems early in the system design process. This included requirements and aims of the system, a list of functions, target group specification and an analysis of the system context and typical scenarios. We conducted several usability engineering loops after the initial benchmarking described in Section 2:

User experience tests: We did partly structured interviews with 52 test users fitting our target group profile. Our aim was to collect the requirements, expectations, wishes and ideas of the users. The test users were members of DAI-Labor and Deutsche Telekom Laboratories. After introducing the system idea to the user the interviewer did a face to face interview. Those interviews took place at the users workplace', because this is the location where they would use the spree system. A realistic location helps to imagine the interaction with the system in the working context.

Paper prototyping: Five users that had previously been exposed to concepts of usability, software development or web applications were involved in paper prototyping. A paper prototype is a low fidelity prototype made of low cost material like colored paper, scissors, tape and pen. This method generates a "quick and dirty" sketch of the structure of the website. The aim is to find an adequate structure of content, navigation and functions test users were given basic structure of the screen and a list of functions the system should offer. Time was limited to one hour.

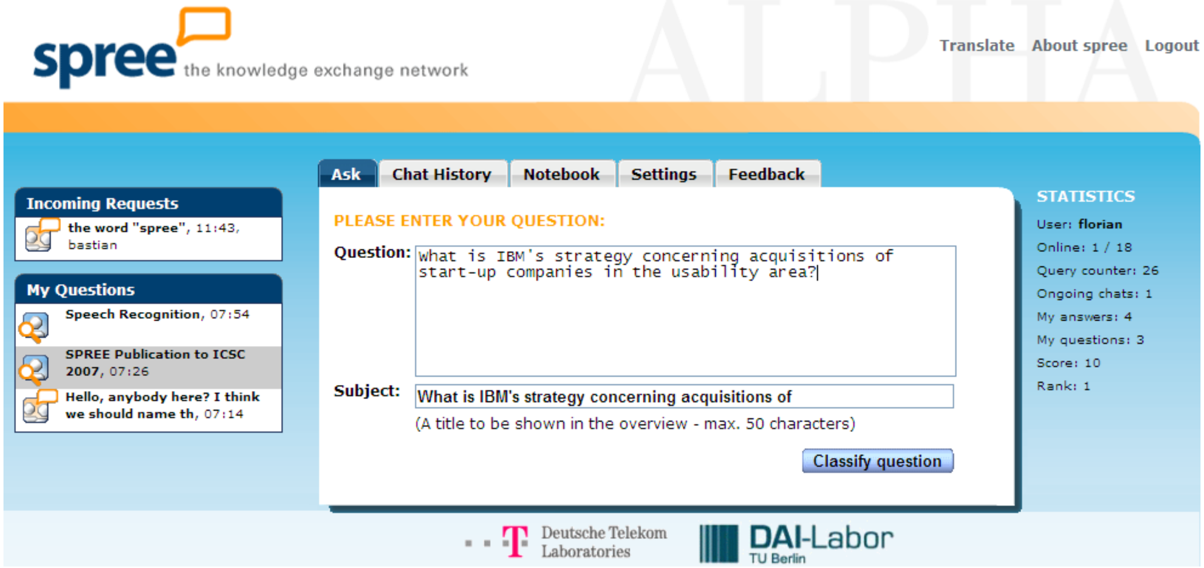

Figure 2. Screenshot of the spree prototype (web search interface) during query submission. The "Subject" field is filled automatically, or can be edited manually. The system will process the question after the user presses "Classify question".

Expert analysis as friendly user test: The aim of this test was to learn about the first interaction of a user with the system. Usability experts played the role of friendly users. The test had an initial interview part about first impressions, then the users should solve some scenario tasks to check the usability of the functions. Overall impressions and experiences with the system were asked as well as demographic data.

Expert discussions and iterative prototype surveying: In the project team we cultivate a lively discussion culture about design topics. Playful explorative surveying of the current state of the prototype implementation is done in parallel to the overall development process.

Online group user test: We are currently planning a user test with several test users at the same time to observe how they interact with each other through the system.

In the late development phase when the prototype is completely implemented we are going to evaluate the different versions of the interface. We plan to investigate interdependence effects between the way a query is formulated (either as a normal text-based web-search as proposed in [11] or as a query in a human-to-human chat) and the result of the matching algorithm.

5.4. Initial Results

We identified critical factors from the users' point of view and developed mitigation strategies, prioritizing according to estimated benefit, feasibility, and cost.

Given the system idea and initial design, users were most concerned about the following topics: no time to answer or chat; little motivation to use the system; unclear social (hierarchical) differences between users; low quality of answers; expert and knowledge not available; ab-use and missing trust for answering; misunderstandings and conflicts in chat. Currently, the system allows users to stay anonymous, only displaying a "nick-name", which users are free to choose at their liking. Users' attitudes to this issue are inconsistent, weighting privacy arguments against trust.

The following features were proposed by users:

User experience tests were conducted with development versions of the interface.

- If an expert's answer was insufficient, ask the same question again, excluding the original expert from the search.
- RSS feed to get automatic alerts on updates to query status (e.g. an expert accepted) or chats, while continuing normal work in the foreground.
- Status (online or offline) of users is shown as in messengers, to see how many users are available.

Users also asked to be able to see a list of all questions in the system, thereby going back to a system without algorithmic matching, because they felt the community would help them learn about new things. Text document and picture upload functions to be able to post richer queries were also requested, as well as closed groups for privacy within working groups.

Paper prototyping showed different solutions to the presentation of questions and answers (messages). Three types were analyzed: 1. all messages are listed in a placeholder, 2. messages are separated in incoming and outgoing boxes, a status bar signals arrival of new messages, and 3. a combination of placeholder and status bar. We first implemented type 1, because users can view the details of a question with only one click, but more research is needed on the best strategy to alert users to the status of their messages.

Expert analysis as friendly user test: given that users interact both with an automatic matching engine and with other Humans in a chat, giving the right feedback to the user is a major problem. Starting with a status bar, we are currently developing more general approaches in order to either make the distinction explicit (web search interface, alternative interface), or make it disappear (chat interface, see Section 5.2).

6. Summary and Outlook

This paper presented a prototype of a web-based expert finding system using machine learning methods in order to identify the best experts to answer a given query, contact them, and establish an online chat between the questioner and an available expert. We presented the main design issues mentioned by potential users of such a system during usability tests conducted in parallel to the development of an algorithm suitable for the task and the implementation of the backend software needed for the spree system. We also presented an initial evaluation of our algorithm and the prototype implementing it.

Tests show that care has to be taken in order to make it clear to users when the data they enter is to be processed by the machine (in order to perform the matching) and when they are already chatting with another Human. We present our ideas on the future design of the interface in order to make this tool available on any Web browser without the need to install non-standard software at the client side, yet allow for efficient communication with the spree server as well as other users.

Our next steps will consist of evaluating the matching algorithm on several tasks and comparing the influence of the matching algorithm and different user interface designs and approaches on the overall performance of the system.

7. Acknowledgments

The authors would like to acknowledge ideas and contributions by Bastian Prillwitz, Robert Wetzker, Sachin Agarwal, Winfried Umbrath, Milena Ilic, and Sahin Albayrak.

References

[1] T. Alpcan, C. Bauckhage, and S. Agarwal. An Efficient Ontology-Based Expert Peering System. In *Proc. IAPR Workshop on Graph-based Representations*, Alicante; Spain, June 2007.

[2] C. Bauckhage, T. Alpcan, S. Agarwal, F. Metze, R. Wetzker, M. Ilic, and S. Albayrak. An intelligent knowledge sharing system for web communities. In *Proc. of the 2007 IEEE International Conference on Systems, Man and Cybernetics*, Montreal; Canada, Oct. 2007. IEEE.

[3] H.-J. Bullinger and S. Hermann, editors. *Wettbewerbsfaktor Kreativität*, chapter Vom Umgang mit der Zukunftsressource Wissen, page 127. Gabler Verlag, Wiesbaden, 2000. In German.

[4] F. Heeren and W. Sihn. Xpertfinder - message analysis for the recommendation of contact persons within defined topics. In *Proc. of the 6th IEEE AFRICON*, 2002.

[5] Human Ressources. Why knowledge is power, May 2007. http://www.humanresourcesmagazine.com.au/articles/81/-0C04C281.asp?Type=60&Category=1765.

[6] International Organization for Standardization. ISO 9241-10 – Ergonomic requirements for office work with visual display terminals, part 10: Dialogue principles. Draft International Standard, 1994. Geneva, Switzerland.

[7] F. Kaiser, H. Schwarz, and M. Jakob. Finding experts on the web. In *Proc. WEBIST 2006*, 2006.

[8] C. McMahon, A. Lowe, and S. Culley. Knowledge management in engineering design: personalization and codification. *Journal of Engineering Design*, 15(4):307–325, Aug. 2004.

[9] J. Moody. The structure of a social science collaboration network: Disciplinary cohesion from 1963 to 1999. *American Sociological Review*, 69:213–238, Apr. 2004.

[10] J. Musser and T. O'Reilly. Web 2.0 – Principles and Best Practices. Technical report, O'Reilly Radar, Nov. 2006.

[11] J. Nielsen. Mental models for search are getting firmer. Jakob Nielsen's Alertbox, May 2005. http://www.useit.com/alertbox/20050509.html.

[12] J. Nielsen and T. K. Landauer. A mathematical model of the finding of usability problems. In *Proc. InterCHI*, pages 206–213, Amsterdam; Netherlands, Apr. 1993. ACM SIGCHI.

[13] M. Ramm, K. Dangoor, and G. Sayfan. *Rapid Web Applications with TurboGears*. Prentice Hall, 2006.

[14] G. Salton. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[15] C. Wu and V. M. Potdar. An overview of preliminary digital ecosystems (des) design. In *IEEE International Conference on Digital Ecosystems and Technologies (DEST 2007)*. IEEE, Feb. 2007. http://www.ieee-dest.curtin.edu.au/2007/Vidy.pdf.