Supporting Collaborative Learning by Matching Human Actors

Andreas Becks
Fraunhofer Institute for Applied Information Technology (FhG-FIT)
andreas.becks@fit.fraunhofer.de

Tim Reichling
Fraunhofer Institute for Applied Information Technology (FhG-FIT)
tim.reichling@fit.fraunhofer.de

Volker Wulf
Fraunhofer Institute for Applied Information Technology (FhG-FIT)
volker.wulf@fit.fraunhofer.de

Abstract

Learning platform focus often on content presentation. Collaborative aspects are mostly dealt with by providing functionality to annotate parts of the content or discuss with other learners about the content. Nowadays learning platforms do not support systematically match making processes among those actors who are able to support their individual learning process mutually. We assume that next generation learning platforms will include functionality to make co-learners aware of each other, match learners with complementary competencies, and allow for the generation of expertise maps. Design principles, a general architecture, and a system providing these functionalities are presented. Future challenges in the field of expertise matching are discussed.

1. Introduction

In this paper we want to focus on match making algorithms to foster collaborative learning processes. Cohen and Prusak [2] argue that networks of informal relationships offer an interesting new perspective to look at knowledge management. While earlier approaches focused on storing and retrieving explicit knowledge represented in documents, new work deals with implicit knowledge ("knowing" according to Polanyi), as well. So, research has to be centered around problem-solving capabilities of individual actors or social entities (e.g. communities). Contrary to the discussion in knowledge management, theories of learning have focused on institutional settings whose primary purpose is knowledge transfer (e.g. schools or universities). Within the field of learning theories, socio-cultural approaches which focus on knowledge acquisition within communities of practice gain in importance [7, 14]. They complement or ever replace approaches which focus on individual learning (e.g. behaviorism or cognitivism).

Given the importance of informal personal networks for knowledge management and learning, applied computer science needs to take this perspective into account. One way to do this is to investigate how computer applications may foster personal networks of learners. In this sense algorithms may make learners aware of each other actors, who have similar or complementary backgrounds, interests, or needs. The challenge is to make potentially fitting actors aware of each other in the virtual or in the real space. So these functionalities offer opportunities to introduce actors towards each other by matching or visualizing aspects of their behavior, background, qualification, expertise, or interests. Therefore these functionalities need to grasp, model and evaluate relevant personal data. These data can be either put in manually by the users, they can be automatically grasped, or they can be imported from other applications.

After reviewing the current state of the art, we will present an application which is supposed to foster personal networks within an e-learning platform. E-learning platforms allow users to access content in a structured way, typically by means of a navigation hierarchy. When grasping personal data automatically, the hierarchical content structure of the e-learning application eases the adding of semantics.

2. State of the Art in Expertise Matching

Research in the field of Computer Supported Cooperative Work (CSCW) and Artificial Intelligence (AI) has created applications which can be a helpful support for matching actors. Expertise profiling systems make explicit and implicit knowledge held by individuals visible and accessible to others. In the standard approach to personal profiling systems the actors are asked to input the data describing their expertise or interests by themselves (e.g. yellow pages). However, the creation and maintenance of personal profiles suffer from a couple of difficult problems. First, a common understanding of the different attributes of a personal profile has to be given [3]. If the profiles are created and updated manually, the different human actors need to have a joint understanding of each attribute. Only in this case their input can be matched automatically. Second, the actors need to be motivated to input and update their personal profiles. Especially the ongoing necessity to update these profiles, threatens their validity [10]. Therefore, these data may be complemented by automatically generated data, derived for instance from analyzing latest changes of an actors' home page or topical mail traffic. However, automatically generated profiles aggregate data of unclear semantics. So it is doubtful whether these data really represent the actors' competencies and interests.

Recommender systems support actors in selecting an item from a set of rather similar items. Several recommender systems are relevant here because they have been designed to support the finding of human actors. Systems like Who Knows [11] or Yenta [4] extract personal data about human interests automatically from documents which are created by the actors. Vivacque and Lieberman [13] have developed a system which extracts personal data concerning a programmer's skill from the Java code the programmer has produced. Based on these personal data the systems allow to pose queries or to match actors. However, these systems are upto now rather dealing with a specific matching algorithms for one type of personal data. McDonald [8, 9] develops a framework of an expertise recommendation system that finds people who are likely to have expertise in a specific area. Contrary to the general approaches to expertise matching mentioned above, the framework allows to develop very specific heuristics tailored to the individual organizational context. So it does not focus on an automatic evaluation of many different documents or programs, but on a context specific heuristic. These heuristics need to be revealed by a preceding ethnographic study in the field of application. If found, such a heuristic is probably better suited than an automatic algorithms. Like in the approaches mentioned above the heuristic matches experts with people looking for support.

3. Matching Personal Data with Algorithms

As the discussion above has shown, identifying, collecting and maintaining appropriate user personal data is very difficult. The information kept in a profile may stem from many different sources (e.g. manually created interest statements, professional or personal history). Moreover, a great deal of semantic background knowledge may be necessary to realize an algorithmic match-making of experts based on personal data.

With regard to the creation of personal profiles in learning platforms, we are in a rather advantageous position. The navigation hierarchy, the course structure, or even ontologies describing the learning content can be used to annotate automatically recorded data. Specific features of a learning platform, like the results of tests, may allow to update personal profiles automatically. So, applications for expertise matching within learning platforms deal with an easier than the general problem.

In this paper we want to use histories of interaction and awareness data concerning the production and use of the platform's content to create and update personal profiles. Due to the fact that the content is pre-structured, an automatically capturing and processing of these data seems to be promising. In the following we want to show which data are relevant and how to gain semantic information from these data:

- data concerning the production of learning material: actors who have produced specific content for the platform may be experts in this domain,
- data concerning the update of learning material: actors who have updated or refined specific content for the platform may be experts in this domain,
- data concerning tutoring responsibilities: actors who are doing or have done tutoring tasks concerning specific content of the platform may be experts in this domain,
- data concerning test results: actors who have passed tests concerning specific content of the platform may be knowledgeable in this domain,
- data concerning the actual use of certain material or the history of interaction: actors who are or have been navigating through specific content of the platform may be interested or even knowledgeable in this domain.

Further aspects of the user's profile can be imported from sources outside the learning platform. Keyword vectors or higher order structures derived from an actor's mail (incoming or outgoing) or document production (letters, papers, slides) can be automatically captured as well as those derived from an actor's homepage [4, 11]. Further data may be extracted from an automatic evaluation of aspects of actors' task performance (e.g. elements of a programming language used [13]). This automatically captured data can be supplemented by profile data entered by the user concerning his personal background, interests, or competences. A cross check between manually and automatically created profile data may reveal inconsistencies. These inconsistencies can be indicated at the user interface to initiate an update of the personal profile.

Now the question arises how to apply these data in matching learners, tutors and content provider. The matching algorithms make use of the ontology given by the hierarchical structure of the content. Whenever a learner looks for support it can be located whenever he browses a specific learning unit. With regard to this learning unit, the system can retrieve data about the production history of the content. The creator of a unit as well as the actor who did the last update can be presented to the learner as well as the one responsible for tutoring. In a similar way histories of passed tests can be applied to match learners with those who have already demonstrated capabilities within a certain time span. Finally the matching algorithms allow to identify those actors which are actually browsing the same learning unit or have done so within a certain period of time.

Prior to presenting our approach to expert finding it is necessary to discuss some further requirements for the matching framework. First, we sketch a direct consequence of the discussion of relevant information sources for user profiles (e.g. information on the professional training status, information gained from produced documents, user context and history of interaction, etc.): Quite clear, each source of information requires a specific method of matching. Thus, the expert matching framework should consist of modules (with well-defined technical interfaces) that encapsulate a certain information type and then contribute to a global matching result by calculating a degree of matching based on that particular type of data (e.g. similarity in learning or project history, interest profile, etc.). Note that this kind of modularity is a pre-requisite for adapting an expert finding component to different application contexts, e.g. different learning platforms or knowledge management environments: Specific matching modules can be exchanged or adapted according to the relevant requirements.

Second, matching expertise affects privacy issues: Learners – or more general: users of any kind of platform that includes expert finding functionalities – might not be willing to make available any kind of personal information to the public. In order to protect the users' right of informational self-determination each user must know and be able to define which of his personal data is used for matching or publication, respectively.

An important problem of matching personal data is the question of 'information quantity': Histories of interaction, e.g., are collected successively and tend to become more expressive with each history item collected. The completeness of personal data may also vary from user to user due to individual privacy decisions. However, the matching quality will depend on the 'completeness' of information available. The more complete a specific type of personal data, the more reliable one can expect the matching result to be. As a consequence for the algorithmic framework a 'degree of completeness' of the different types of personal data should be measured where possible and be taken into account when calculating a matching degree.

Finally, given that a user agrees to use certain types of personal data for the matching process, he should also be able to adapt the expert matching algorithm to a certain degree: The perception of which modules, i.e. types of personal data, contribute to a good profile matching may vary for different users in different contexts. In order to let the user decide which 'factors' contribute to which degree to the expert matching, we propose to incorporate a factor which weighs the impact a certain module has on the overall matching result into the matching framework.

4. A Modular and Adaptable Expert Matching Approach

This section presents an algorithmic framework for expert matching that takes into account the requirements discussed above. In the following we make use of the following terminology: Expert finding means matching a prototype set of personal data (i.e. profile of a certain user in the application environment or a query profile) against a collection of other actors' personal profiles in order to determine a ranking of fitting actors. We distinguish between two modes of using an expert finding component:

- **Filter functionality**: In this mode a user applies the expert finder system in order to find other users with personal data that are similar to his own (or relevant parts of his own data, respectively). This functionality is not only relevant in a learning environment where a learner wants to find other learners with similar backgrounds, interests and knowledge in order to build a learning group. It also applies to organizational knowledge management scenarios: Consider an enterprise environment where an expert in a certain field wants to set up an expert network of people with similar project background in order to share experiences. Similarly to filtering profiles against one's own profile the user can pose a query (in terms of a user-defined profile) to the expert finder system in order to find people that match the explicit needs of the user. As an example consider an enterprise environment where an employee needs to find an expert in a certain field who is able to solve a specific problem.

- **Cluster functionality**: Here, the expert finder system is used to cluster the profiles of all users in order to present a "landscape of expertise" for analysis or exploration purposes. Consider an enterprise environment where a project manager tries to identify the expertise of those members of the staff who could potentially take over a certain subtask.

The term "personal data" is deliberately kept very general: With regard to the discussion in section 3 data subsets may include an actor's professional and training status, interest statements, self-assessment of abilities, certain kinds of history information (learning history in a learning environment, project history in a company's expert database, etc.) and similar data that describes the user's expertise depending on the application context.

We now describe the algorithmic matching framework of our expert finder system in a more formal way. We point out the elements for making matches of profiles and define their constraints. Let P denote the set of all possible personal data (i.e. user profiles or query profiles). These data consist of the full set of information available for each user or the query, respectively, in a certain application environment (i.e. a certain learning platform, an enterprise expert database, …).

As discussed above, there are different kinds of data that help to determine the degree of expertise. According to the nature of the data subsets, different algorithms for matching subsets (e.g. self-assessment, history) have to be developed and applied. In the following this is expressed by the notion of modules: Each module contains functions for matching the respective relevant subsets of personal data (which contain the data items used for matching). Intuitively, each module realizes a criterion for expert matching (e.g. one module for matching histories of interaction, another for matching the training status of users).

Formally, each module M_i, $1 \leq i \leq k$, consists of a **matching function** $m_i : P \times P \rightarrow [0,1]$ which determines the degree of similarity1 for each pair (p_a,p_b) of personal data collections (e.g. the similarity of learning histories) and a **completeness function** $c_i : P \rightarrow [0,1]$ which measures the degree to which relevant data for M_i is available in a profile p (e.g. the amount of interaction history in the profile, see also section 3).

The matching function is realized for each module. The realization of this function has to meet the following requirements: The more similar two profiles are, the higher the value calculated by m_i (with 1 representing a 'perfect' match). Furthermore, matching identical profiles should produce a perfect match (which is quite intuitive), i.e. $m_i(p_a,p_a) = 1$ for all $p_a \in P$. In cases where the clustering functionality is used we also assume m_i to be symmetric, i.e. $m_i(p_a,p_b) = m_i(p_b,p_a)$ for all $p_a, p_b \in P$.

The completeness function takes into account the quantity of data available for module M_i's matching process, where $c_i(p) = 0$ means that no data is available and $c_i(p) = 1$ corresponds to the maximal degree of completeness. For example, assume a module M_i that matches profiles based on certain history data of a user. If the user is new to the platform in which the expert finder is embedded only few (or even no) history data may be available in p. In this case the matching function m_i might yield a high matching value based on the considered pair of profiles, but this matching would be based on sparse data. In such a case $c_i(p)$ should yield a low value[2].

In order to allow the user to adapt the matching process he can adjust the influence the different matching criteria have on the overall matching result. Formally, this is done by assigning weights to the single modules: For each module M_i a weight $w_i \in [0,1]$ is given where $w_i = 0$ means that module M_i is switched off and $w_i = 1$ means that module M_i has full influence. All values of w_i between 0 and 1 correspond to a more or less strong influence of module M_i.

We also need to take care of privacy issues: If a user does not want the system to use certain parts of his profile for match-making he should be able to switch off the corresponding modules. Formally, we introduce flags $priv_{ai}, priv_{bi} \in \{0,1\}$ for all users a and b and each module M_i which indicate whether the respective user wants module M_i to be used or not.

Now we can define the overall matching result for profiles p_a and p_b from users a and b for all modules M_i, $1 \leq i \leq k$:

$$m(p_a,p_b) =_{def} \frac{1}{k}\sum_{i=1}^{k} priv_{ai} \cdot priv_{bi} \cdot w_i \cdot \min(c_i(p_a), c_i(p_b)) \cdot m_i(p_a,p_b)$$

Summing up, the overall matching value of two profiles p_a and p_b is based on the matching degree of each individual module. A module can only contribute to the overall result if both, user a and user b, agree that their profile may be used for match-making by setting the respective flag to 1. Furthermore, the contribution of each module depends on the completeness (and thus, as we claim, trustworthiness) of the respective profile data and the user-defined weighting for that module.

5. Realization of the Expert Finder

Based on the methodical approach presented in this paper we built an expert finder system for the Fraunhofer e-Qualification framework – an e-learning environment that offers extensive technical, methodical and didactical support for both, authors of Web-based trainings and learners. We therefore adapt the generic architecture of our system to the e-Qualification platform and provide a user interface as presented in section 5.2 which mainly focuses on the filter functionality of our framework. In section 5.3 we sketch another application of our framework where the actors' personal data are input for the cluster functionality: We present the idea of an advanced visual interface for exploring a "landscape of expertise".

1 Note that dissimilarity (‚distance') measures can be used as well since their results can be converted to similarity values.

[2] Of course of certain matching functions may take sparse data into account and yield corresponding matching values. In such a case consider $c_i(p)=1$.

5.1. Architecture

The architecture (cf. figure 1) we have chosen for our expert finding system has been designed in order to keep the system flexible and easily adaptable to different application platforms. At this point it is important to properly distinguish between expert finding-system, the ExpertFinder and the ExpertFinder-modules. In figure 1 the expert finding-system is located in the bottom left surrounded by the dashed line. It consists of three major parts: The expert finder itself (which realizes the algorithmic matching framework described above), the connection to the application environment (e.g. a learning platform) and the connection to the client (i.e. Web browser). Here is a description of each of these parts in more detail.

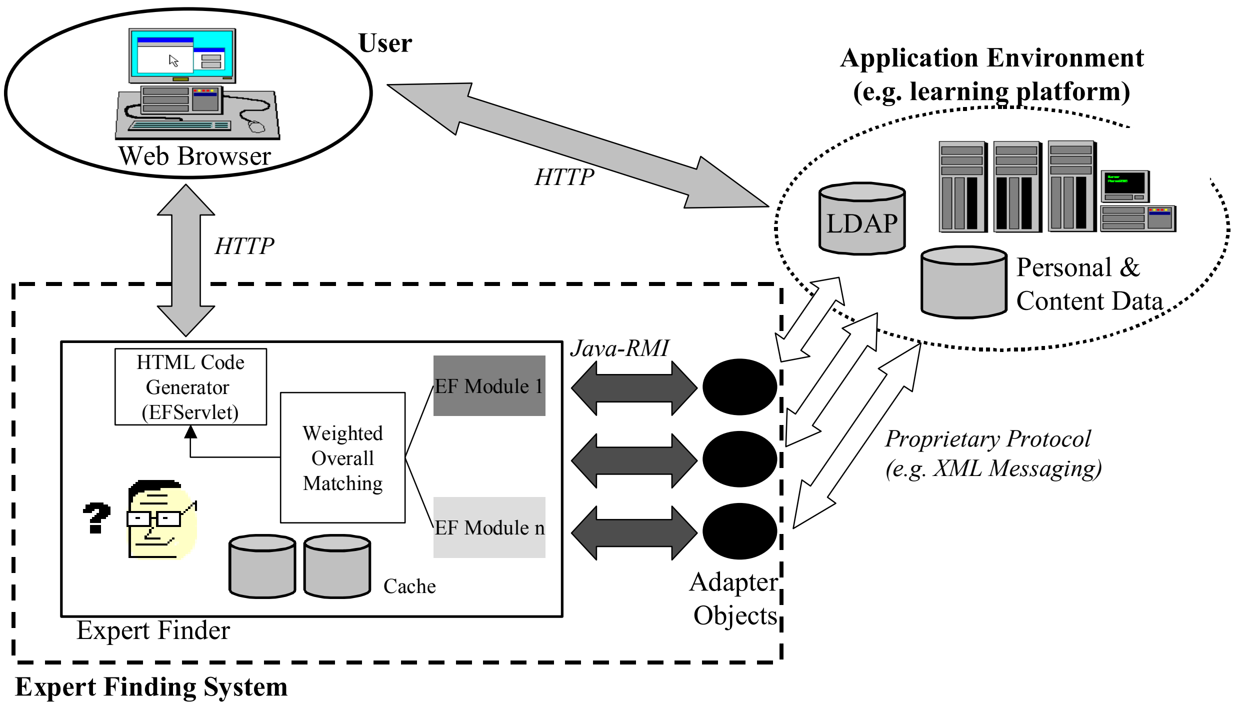

Figure 1: Architecture of the expert finding system

- First, as we wanted to keep the system adaptable to different application platforms, we couldn't expect a uniform way of data exchange with the application environment. So we had to invent adapter objects that could be easily and quickly developed. These adapter objects had to translate platform-specific data into a format that the expert finder can deal with. As figure 1 shows, there are three of them: One to have access to the user-database (in case of the e-QF-framework this is an LDAP-directory), another to have access to the content-database (in case of the e-QF-framework the content is stored in a tree-like structure, where the leafs are the WBTs and the internal nodes are html-based pages leading the user to the desired content). Regarding the third adapter object it is important to understand that the expert finder does not only rely on data that is given by user-profiles but on user-history-data. Thus it is necessary to keep the system aware of the user actions. This is done by a push-event that is invoked by application platform to 'tell' the expert finder that a certain user has moved from one content to another – a potentially a valuable information on the user's current interests. The third adapter exists to trigger the 'user moves'-event and to translate the parameters (which user moves to which content).

- The connection to the client is done by a Java™ servlet that interacts with the expert finder and generates the (html-based) user interface.

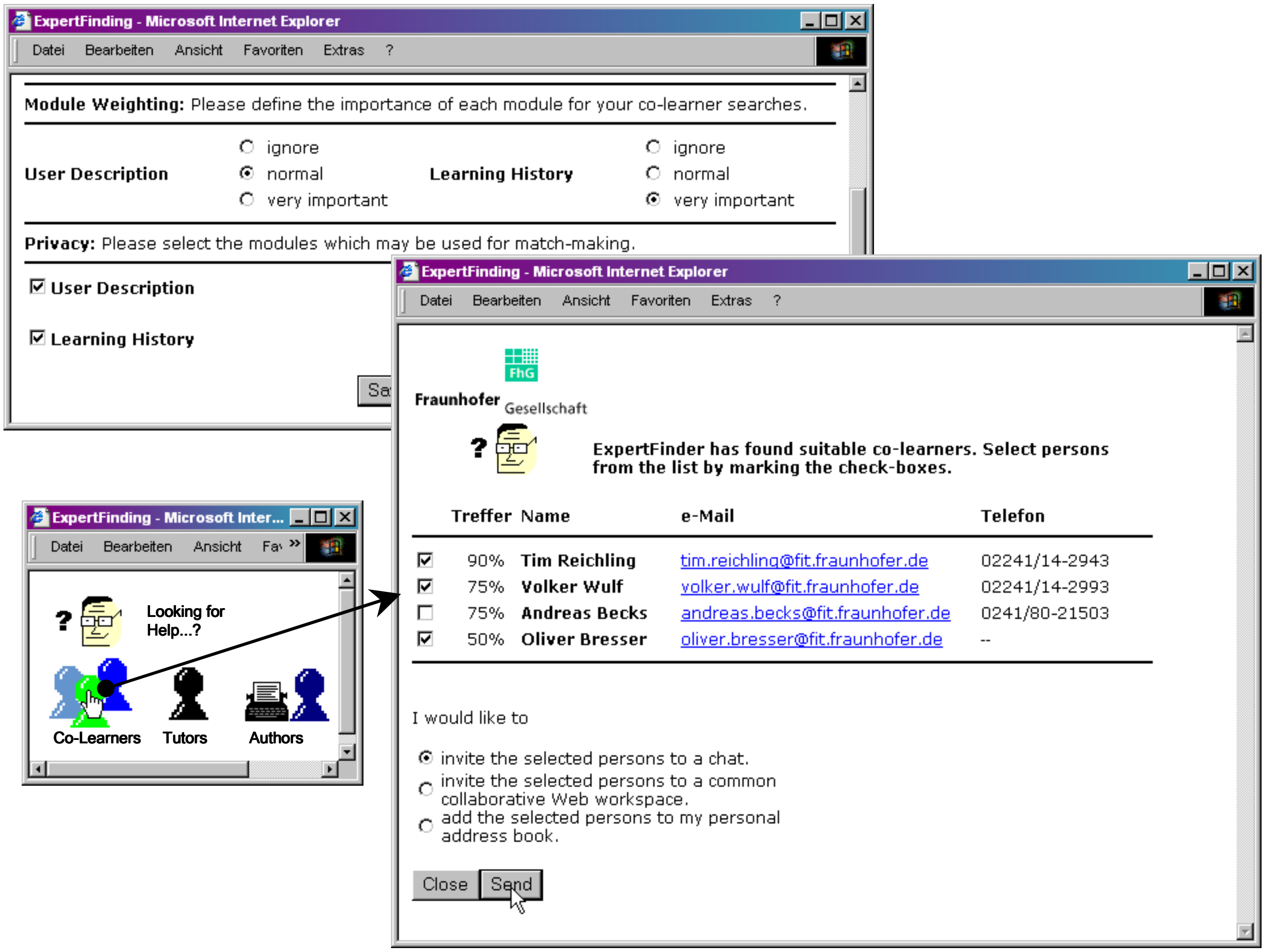

Figure 2: The expert finder interface of the e-Qualification platform: main window of the expert finder (bottom left), result list of co-learner search (bottom right), options window for configuring modules for co-learner search (top left)

- The main component of course is the expert finder itself. It contains internal databases where information about the users and the content of the application environment is 'cached'. This is done to have quick access to those information that are frequently used by the expert finder, and – this is the main goal – to store additional data about users and content which is not kept in the main-databases of the application environment. The main task of the expert finding (i.e. matching users to each other) is left to the expert finding modules, each implementing a different matching strategy. When one of the above mentioned 'user-moves'-events occurs, this event is forwarded to all the expert finder-modules that are currently running. Each of the modules now can decide whether this event is important with respect to the strategy that it implements and thus has to be stored as history data. Finally the 'scores' of the modules are collected and weighted by the weighted overall matching to have the results in an order according to the users preferences (one user might prefer scores that aroused by a matching strategy based on user-profiles, another preferred strategies based on history data).

5.2. User Interface

Figure 2 depicts the user interface of the expert finder within the Fraunhofer e-Qualification platform: Users of the platform who are logged in as learners can use the expert finder system in order to contact authors, tutors and suitable co-learners. For learners, the main window of the expert finder is available during the complete training session (either as a separate window or integrated in the training web pages as a frame layout). Learners can seek advice from the author of their whole session or specifically assigned tutors by clicking on the respective symbol in the main window. In both cases the result of the user request is the name, e-mail address and telephone number of the author(s) or tutor(s), respectively.

When the learner looks for potential co-learners he can click on the corresponding icon in the main window (or expert finder frame). As a result of this request the learner directly receives a list of suitable co-learners, ranked by the matching degree of his own profile and the profile of each candidate co-learner (thus realizing the filter functionality of our framework, cf. section 4). The user can then select potential co-learners from the list and either invite them to a chat (as a means of synchronous communication), ask them to join a common collaborative workspace within the learning platform (as a means of asynchronous communication), or just add tem to his personal address book for later reference.

In a specific options window the user can define a relative weighting of the modules used to assess the suitability of other users of the platform as co-learners, thus determining the influence each module has on the matching result. In this case the modules “learning history” and “user description” are available. The latter contains, among others, educational status and interest statements of each learner subscribed to the platform. The options window also allows to adjust the learner’s individual privacy flags. The privacy flags are used globally, i.e. as soon as a learner deactivates a flag, the corresponding profile data will not be used by the expert finder at all. The module weights, on the other hand, are used locally: If a learner selects “ignore” for one of the modules, this only affects his co-learner searches. For the searches of other learners the respective personal data is still available.

5.3. Expert Maps: Clustering Profiles for Exploration

In this section we sketch another application of our expert finder framework where the calculated similarity information of personal profiles is used to automatically calculate a graphical “landscape of expertise”: In larger traditional enterprises or in virtual organizations similar or complementary expertise is usually spread over different departments or sites, making it difficult to get an overview of the areas of competence available in the organization. We conducted informal interviews in several medium-sized companies which were interested in advanced knowledge management technology. The study revealed that a graphical presentation of an “expertise landscape” would be of high value. We therefore propose to combine the clustering functionality of our framework with a technique for visualizing the structure of complex information spaces, originally developed for data and document mining [1]. Given a degree of similarity for each pair of actors (e.g. the similarity of each pair of personal profiles), the visualization approach automatically generates an overview map (cf. figure 3) which intuitively displays the relationships of actors are represented as points in the map; similar collections of personal data are grouped as neighbored points – the more similar, the closer they appear in the display. In addition, the shades of gray indicate additional similarity information: Groups of similar actors can be found in the bright-shaded areas. Dark borders separate these groups. The darker the border, the more dissimilar are the respective profile groups.

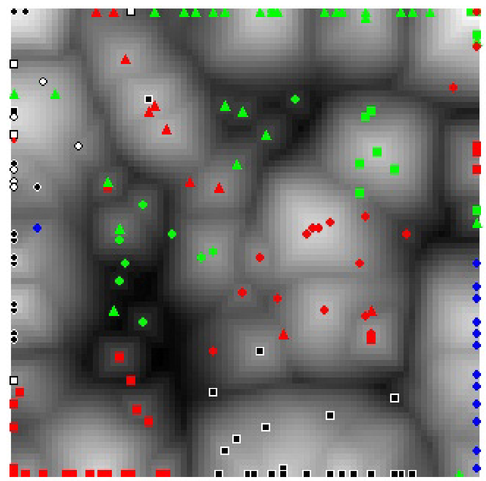

Figure 3: An expert map which displays the similarity structure of all pairs of personal profiles can be used for navigating through and exploring an “expert landscape”.

The map in figure 3 shows how a landscape of expertise based on personal data of employees in a medium-sized company could look like. The color and shape of profile points in the map represent the department or site the respective employee (who is represented by the personal profile) works in. Applications of such expert maps (as discussed with our industry partners) would include the identification of competence areas and their distribution within the company or the identification of only sparsely covered areas of expertise. In other applications project managers would use the map for identifying suitable team members by browsing through the competence landscape, and innovation managers would apply the map for identifying groups of people with related or complementary personal profiles in order to set up networks of experts for experience exchange and creative brainstorming sessions. Initial discussions of the concept with our industry partners yielded positive feedback.

6. Empirical Investigations

Empirical explorations and evaluations are essential to develop our approach. To work out the algorithmic matching framework, three type of activities are needed.

- empirical evaluations to validate the existing algorithms for matching modules,
- empirical explorations to create further algorithms for matching modules,
- empirical evaluations to validate the adaptive match-making framework.

Right now two different modules for matchmaking are realized. The first module matches co-learners according to their histories of interaction with the learning material. It is described in section 5. The second module matches learners according to their educational and professional background. Data concerning the school degrees, professional domain, and working or management experience are provided by the learning platform. The algorithms matches learners with similar backgrounds. To evaluate this specific module, we plan to apply the match-making strategy to the members of our research institute (about 150 researchers and students). After confronting our colleagues with the outcome of the match-making, we will conduct semi-structured interviews to find out whether the suggested partners are regarded to be an appropriate choice. Moreover, we will try to find out which additional data concerning educational or professional background should be taken into account for match-making.

To explore new ideas for match-making we will conduct a series of open interviews with people who learn or solve complex problems together in highly dynamic environments (e.g. virtual organizations). We will try to identify factors which people find relevant in cooperating with their partners. Based on these findings, we try to find data which may acts as an electronic surrogate for the factors identified. Finally, we will have to evaluate whether the adaptive match-making framework is appropriately designed. We will have to find out whether the users are able to understand an actual state of configuration and whether they are able to adapt the framework by manipulating the different weights. This aspect of the evaluation refers to the design of the user interface for tailorable applications.

Beyond merely design oriented empirical investigations, we will have to see whether and under which circumstances such match-making algorithms are accepted by users. Key questions are whether users are willing to provide personal data for matching purposes and whether they are willing to establish social contacts based on technically generated recommendations.

7. Conclusion

In this paper we have looked at computer applications to encourage human learning. We assume that dense personal networks among learners will have a positive effect on their performance. Given the fact that actors within e-learning platforms will not know each other well, we have developed a framework for match-making. Contrary to face-to-face learning environments, e-learning environments do not allow to meet in person. So many of the real-world clues which enable the establishment of initial forms of personal relationships among learners are missing (e.g. behavior, gesture, facial expressions, or eye contact). The question comes up whether there can be virtual substitutes which may ease the establishment of social relationships. In this paper we have explored ways to match the learners' personal data. We have developed a framework which allows to match different subsets of personal data in a weighed manner. Depending on the subset of personal data, the matching algorithms may favor similarity or complementarity.

With regard to the development of new generation learning platforms, our research calls for a detailed recording of the learners' interaction with the platform. Moreover, a fine grained exchange of data between the learning material and the learning platform is required (e.g. transmission of test results). Further efforts to standardize this data exchange are required. While we have chosen a stand-alone architecture for the expert finder, one may also want to integrate this functionality into next generation learning platforms. This is of special importance for building adaptive learning platforms (cf. [12]). Algorithms which adapt a platform to the assumed needs of their users need similar personal data on the learners' behavior. So these data should be stored within the platform to be efficiently accessible to adaptive algorithms as well as to match-making algorithms.

Many research issues discussed in this paper need further investigation. Refinements of the different matching algorithms are required. We have to find out which are the right subsets of personal data and which algorithms are best suited to match a certain subset of data. As the application context will have a strong influence on the choice of the appropriate algorithms, the expert finding components should be highly tailorable. Appropriate concepts to tailor these applications have to be developed (cf. [15]).

Further research is required to find out how to support the building of relationships after having matched the actors. Questions arise such as: which private data should become visible to the matched partners, what are the appropriate media for communication, which role may face-to-face meetings play in such an environment? Another research issue refers to the preconditions necessary to successfully match actors. Probably a somehow common language and culture are an important prerequisites to establish contact between the actors. These factors cannot be easily grasped by means of personal data.

We have approached the problem of how to support the establishment of social relationships in the virtual world where many traditional clues for self-selection among the actors are missing. Our approach to expertise matching in an e-learning environment needs empirical evaluation and further refinements. A thorough empirical investigation of potential risks and opportunities is required.

References

[1] Becks, A., S. Sklorz, M. Jarke: Exploring the Semantic Structure of Technical Document Collections: A Cooperative Systems Approach. Fifth IFCIS International Conference on Cooperative Information Systems (CoopIS'2000), Eilat, Israel, September 6-8, 2000

[2] Cohen, D.; Prusak, L.: In Good Company: How Social Capital makes Organizations Work, Harvard Business School Press, Boston 2001

[3] Ehrlich, K.: Locating Expertise: Design Issues for an expertise locator system, in: Ackerman, M.; Pipek, V.; Wulf, V. (eds): Expertise Sharing: Beyond Knowledge Management, MIT-Press, Cambridge MA 2002 (in press)

[4] Foner, L. N.: Yenta: A Multi-Agent, Referral-Based Matchmaking System, in: First International Conference on Autonomous Agents (Agent'97), ACM-Press, New York 1997, pp. 301 - 307

[5] Kautz, H. A.; Selman, B.; Shak, M.: Referral Web: Combining Social Networks and Collaborative filtering, in: Communication of the ACM, Vol. 40, No. 3, pp. 63 - 65

[6] Kautz, H. A.; Selman, B.; Shak, M.: The hidden web, in: AI Maganzine, Summer, 1997, pp. 27 - 36

[7] Lave, J. und E. Wenger: Situated Learning: Legitimate Peripheral Participation. Cambridge University Press, Cambridge 1991.

[8] McDonald, D. W. (2000): Supporting Nuance in Groupware Design: Moving from Naturalistic Expertise Location to Expertise Recommendation, PhD-thesis, University of California, Irvine 2000

[9] McDonald, David W.: Evaluating Expertise Recommendation. Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work, Boulder, Colorado, USA, 2001

[10] Pipek, V.; Hinrichs, J.; Wulf, V. : Sharing Expertise: Challenges for Technical Support, in: Ackerman, M.; Pipek, V.; Wulf, V. (eds): Expertise Sharing: Beyond Knowledge Management, MIT-Press, Cambridge MA 2002 (in press)

[11] Streeter, L. A.; Lochman, K. A.: An expert/expert location system based on an automatic representation of semantic structure, in Proceedings of the Forth Conference on Artificial Intelligence Applications, San Diego, CA, March 14 - 18, 1988, ACM-Press, New York 1988, pp. 345 - 350

[12] Specht, M.: Adaptive Methoden in computer-basierten Lehr-/Lernsystemen, GMD Research Series, St. Augustin 1998

[13] Vivacque, A.; Lieberman, H.: Agents to assist in finding help, in: Proceedings in the Conference on Computer Human Interaction (CHI 2000), ACM-Press, New York 2000, pp. 65 - 72

[14] Wenger, E.: Communities of Practice, Cambridge University Press, Cambridge1998

[15] Wulf, V.: Zur anpassbaren Gestaltung von Groupware: Anforderungen, Konzepte, Implementierungen und Evaluationen, GMD Research Series, No. 10/2001, St. Augustin 2001