

Supporting the Formation of Informal Learning Groups in a Heterogeneous Information Environment

Adam GIEMZA*, Sven MANSKE, H. Ulrich HOPPE
University of Duisburg-Essen, Germany
*giemza@collide.info

Abstract: University freshmen have to cope with complex and heterogeneous information infrastructures typically found in nowadays universities. Usually learning management systems like Moodle or Blackboard are applied for lectures. Additionally, Cloud Services like Google Drive, Brainstormer, and Doodle are meanwhile frequently used as tools for learning in various contexts. They support storage, content production and particularly also coordination. The management of these heterogeneous tools is a challenge for the individual users as well as for the usage in groups. This paper presents a mobile application to support the learners in the formation of informal learning groups and integrates heterogeneous cloud services to support group formation and further group work in a campus environment.

Keywords: information ecologies, cloud services, mobile learning, group formation, recommender systems

1. Introduction

Nowadays presence-based university studies are supported by a complex and heterogeneous information infrastructure: Course-specific information is typically embedded in learning management systems, with possibly different systems used even in one university. Administrative and organizational information about study programs, general student services etc. is often provided through other information channels than the course-specific information. Students access, partially store and further manage this information on personal devices such as notebooks or smartphones. Generally available social media and cloud services may be used to share and further distribute such information. In this sense, a modern campus information environment is an example of a complex, heterogeneous and somewhat scattered infrastructure (Hanseth, 2010).

From the given characterization we can directly construe the challenge of better integration. Integration can be addressed from two sides: from the sources (i.e. information and service providers) or from the user (student) perspective. Integration from the source would need to be based on strong premises of being able to change large parts of grown systems. So, in a pragmatic approach we address the problem from the user perspective.

We have specialized the problem for a typical application scenario: Especially students in their first semesters on campus face the challenge of finding peer groups to collaboratively work on assignments or prepare for exams. This includes the problem of group formation and basic support for group work. As a further specification, we have set a focus on delivering and managing information on personal mobile devices and on considering opportunities in space and time in a campus environment. This conceptual and technical challenge has been addressed in a software development project with master level students from an interdisciplinary study program on interactive media and applied cognitive science.

2. Related Work

Providing orientation support for freshmen on campus has been a theme of a number of mobile learning applications (Giemza, Verheyen, & Hoppe, 2012; Lucke, 2011), often also using a game-based approach. While these approaches focus on learning the university campus and the

surroundings, our application focuses on getting to know new fellow students to form learning groups. The gamification aspect has a lower importance, as we believe that the students have an intrinsic motivation (Ryan & Deci, 2000) to find learning groups for gaining knowledge from others in the collaborative learning process.

Jansen et al. (2005) describe a prototypical campus information system that integrates interactive public displays with personal mobile devices and supports personalized and location-aware information transfer. If such an infrastructure (including interactive displays distributed over the campus) were widely available and accessible this would have been an ideal basis for implementing the specific functions supporting group formation and support of group work. However, unfortunately we cannot rely on such premises.

The formation of learning groups has been studied from an intelligent systems perspective using quite sophisticated processing techniques (Hoppe, 1995; Isotani, Inaba, Ikeda, & Mizoguchi, 2009; Largillier & Vassileva, 2012). In our case, we do not rely on deep knowledge modeling and intelligent learner diagnosis as input to the group formation process, although this could be considered in future versions. Our starting point and current focus is on opportunistic usage of simple user profiles, general information and coincidences in time and space.

Jansen et al. (2013) have classified the usage of existing cloud services from an educational perspective. They distinguish different types of services, including archiving/repository services (such as Dropbox), communication and coordination services (such as Twitter or Doodle) and rich production services beyond simple text messages (such as MindMeister). Still, simple repository or communication services are predominant in many educational applications. A future potential is seen in the area of processing services and especially services for learning analytics.

3. Meet2Learn

3.1 General

Meet2Learn aims at supporting students in meeting new fellow students with the goal to form learning groups for their joint lectures. This will not be achieved by a self-contained standalone application but by integration of different and heterogeneous information sources into one environment. Once the learning group has been formed, the system will act as an organizer to integrate learning materials and other artifacts worked out collaboratively from different cloud services.

In Meet2Learn learning groups are informal groupings organized by students themselves, in contrast to exercise groups offered beside a lecture. Here the learners take the initiative to organize and execute the learning group. Nevertheless they are formed around a lecture that builds the context and the content of the learning group. To support learners in creating and finding appropriate learning groups, the system needs to be aware of the lectures being offered at the University. This will prevent the users from creating multiple learning groups concerning one and the same lecture - only due to different naming of the lecture by the learner - while searching for a group or while creating it. Furthermore learners need to be represented in the system to provide grouping support based on a user profile. This profile needs to contain the course of studies, the number of semesters and finally the attended lectures. This will empower the system to provide recommendations based on user-based collaborative filtering methods (Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994). Finally, universities provide various learning management systems (LMS) like Blackboard or Moodle, which can be integrated with the learning groups to import learning material from the lecture as well as to export collaboratively elaborated content back into the LMS. This content will be exported from the “production-type” cloud services like Google Drive or Brainstormer (Brainstormer, 2013).

In the sequel we will explain the components of the Meet2Learn applications in more detail, starting from an architecture overview, the mobile application and finally the recommendation approach. The target groups for the first prototype of Meet2Learn are students at the University of Duisburg-Essen studying *Applied Computer Science* and *Applied Communication and Media Science*. Therefore the system imports the lectures from the university system called LSF and exports learning groups into a Moodle provided by the university as well.

3.2 Architecture Overview

The frontend of Meet2Learn has been developed as a native mobile Android application. This results in the benefit of a practically permanent availability to the students for checking the status of learning groups and receiving notifications about relevant information. The backend uses an agent architecture on top of a blackboard system (Weinbrenner, 2012). This allows for a flexible and extendible design with connection to heterogeneous services using multiple programming languages. The integration of cloud services takes place on the server side through agents as well as directly on the client side through the Android application. The recommendation in contrast is done only on the server side through recommendation agents (see Figure 1).

The application integrates the following types of cloud-based services on the smartphone: *Communication*, *Production* and *Repository*. Twitter and Facebook are integrated as the first type of service. Hereby students can announce learning groups to their friends that are not yet using the Meet2Learn application. This will increase not only the number of users but also the probability that a learning group will take place with enough participants. Furthermore it will increase the extrinsic motivation (Brophy, 2004) to the learner, as he or she announced the intention to participate in a learning group to the public. Google Drive and Brainstormer are integrated as cloud-based production services. Here learners produce learning artifacts collaboratively and link them to the learning group for sharing the results. Dropbox is integrated as a cloud-based repository service. Learners can link uploaded resources and provide direct access through the learning group. The client-based integration makes use of the rich connectivity and extendibility of Android through flexible built-in and third-party APIs.

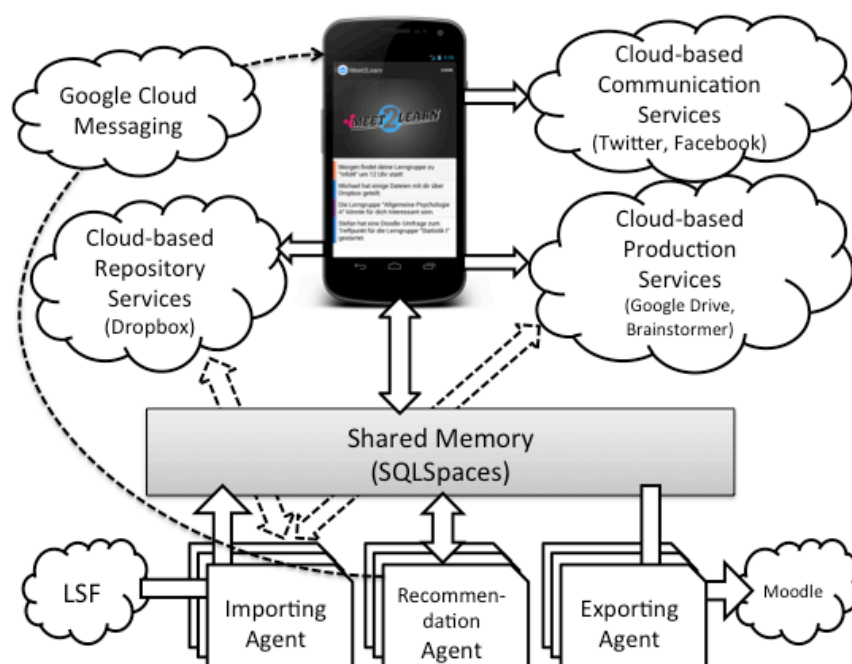


Figure 20. Meet2Learn Architecture Overview

On the server side, we use an agent architecture utilizing the SQLSpaces (Weinbrenner, 2012) as a shared memory with different tasks assigned to agents. In the lower left of Figure 1, importing agents gather data from different heterogeneous sources and import them into the system. The current configuration integrates the study program from the central repository of the university (LSF). This data is used to support the students in adding lectures to their profiles and prevents multiple notations for one lecture. The recommendation agents are responsible for processing user profile data and for generating recommendations. One particular feature is the notification of the learners about new recommendations for a learning group based on the user profile or opportunities (e.g., a new learning group of interest will take place closely to the users' current location). Notifications are implemented using the native Android notification mechanism and the Google Cloud Messaging (GCM) service. The recommendation approach used in Meet2Learn will be explained in more details in section 3.4. Finally, exporting agents integrate learning groups into external systems, like the aforementioned

learning management systems (LMS). We are planning to integrate the university's Moodle by exporting Meet2Learn learning groups into Moodle groups. This will allow learners to continue the work of a well-established learning group inside Moodle closely located to the lecture.

3.3 Mobile Application

The main entry point for the learner to the system is the Meet2Learn mobile application. Figure 2 shows three main views for the user. The first view (a) represents the main entry point to the application. Here the user finds a “news feed” containing the latest information about learning groups as well as notifications with recommendations and reminders to the learner. The messages are typed by a color tag on the left and can be therefore quickly grasped and identified. A click on the message opens a new view presenting more details about the event. This view therefore only visualizes events and the results of the server side recommender system. The second screen (b) shows the group creation screen. The learner can configure a new group by setting multiple parameters like the name, time and place and publish this new group to the system. Fellow students can search this group and join it. Finally, the users are responsible themselves to participate physically in the informal learning group.

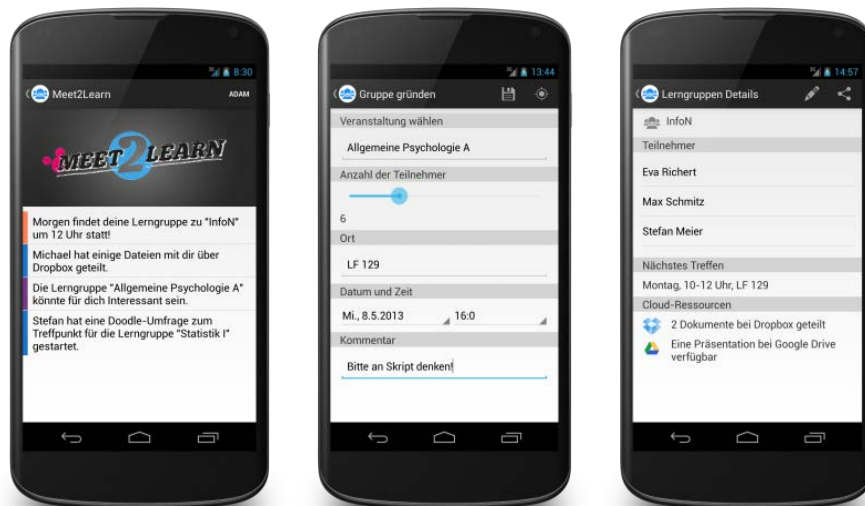


Figure 21. Meet2Learn App - (a) News feed and notifications (b) Group creation view (c) Group details view

The last screen (c) shows the details of a learning group. The learner can see the title, the participants, the next meeting and finally the resources shared within the learning group using the aforementioned cloud services (here Dropbox and Google Drive). For this purpose Meet2Learn registers itself as a recipient of “share events” of other applications on the Android phone. A user can therefore open the Google Drive app and share the link to a resource with the Meet2Learn app. This will prompt the user to select the correct learning group to share the document with. All participants of this group will receive a notification that a new resource has been shared with them in the shared group. The view (c) also implements the standard share button (top right) that allows the users to publish a learning group on cloud-based communication services like Twitter and Facebook. In this example, the user can share the information about the learning group for the lecture “InfoN” by clicking the share button. The system will present a list of possible (cloud) services to use to process the message. If the user selects Twitter, the system will open the common Twitter dialog with a prefilled message about the group and a link to a web page presenting some more details about the learning group (Figure 3). Finally the user can send the tweet and share this to his or her followers. This allows promoting the learning group to other learners not using Meet2Learn and thus allows new users to join the Meet2Learn community.

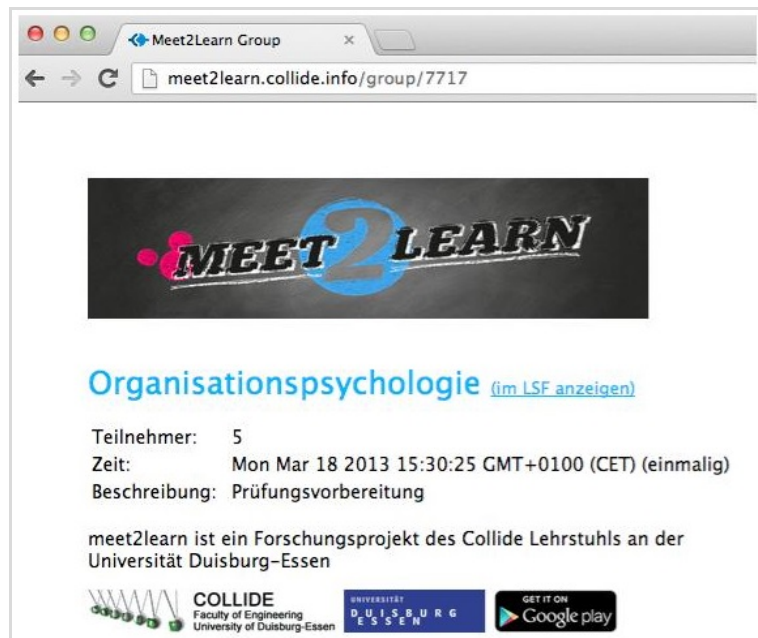


Figure 22: Web page with details about the learning group

3.4 Recommendation Approach

Discovery of learning groups is a crucial aspect of Meet2Learn as it affects the decision whether to create a new learning group or to join an existing one. One possibility for discovering learning groups is an active exploration through searching and browsing groups by the user on the device. To reduce the accompanying risk of information overload that is given through search and especially browsing, recommender systems as information filtering systems provide a passive way to discover new items in a system (Maes & others, 1994). Recommendations for learning groups are generated by the system and presented to the user in the mobile application. In the Meet2Learn system this means predicting the users' interests for a specific course inferred from his/her specific profile compared to the other users' profiles and offering relevant learning groups for this course as recommendations.

In the current version Meet2Learn uses a user based collaborative filtering (CF) approach (Resnick et al., 1994), utilizing the k nearest neighbors of a user to predict her interest in courses. Therefore a meaningful representation of a user in a feature space is created to calculate similarity measures. Each course leads to a feature, where the values capture the user's visited courses, course interests or participation in learning groups. The binary feature vector encodes these user-course adjacencies. A ranked list of recommendations is created through a weighted average of the nearest neighbors based on the inverse Euclidean distance. The user's yet undiscovered courses that will be recommended have a positive, non-zero mean value in the nearest neighbors vectors, reflecting that similar users are interested in this course while the user herself has not explicitly indicated an interest. In this relatively low-dimensional data (around 19 course items for the study course in question) it is very likely to have several users that are too similar, namely having a similarity measure of 1. The consequence is that the system cannot infer any new interests from the neighbors' profiles in such cases. For that reason, k nearest neighbors are considered, with $k > 1$. Increasing the value of k could also affect the stability of the nearest neighbors (Beyer, Goldstein, Ramakrishnan, & Shaft, 1999).

A common challenge for collaborative filtering is the cold start problem (Lam, Vu, Le, & Duong, 2008; Schein, Popescul, Ungar, & Pennock, 2002), which mainly occurs for the system events *new item*, *new user* and *new community*. Items in the sense of Meet2Learn are courses. If a new course is added to a system, no user will receive a recommendation for the course, as there are no relations present. A new user will not be similar to other users, who already visited courses or indicated interests. Therefore no useful recommendations for the new user can be generated by the system. In a new community all users have the same similarity but no indicated preferences, which prevents making inferences about interests. These problems are widely known and usually solved with hybrid recommender systems that combine CF with content based filtering (Burke, 2002; Schein

et al., 2002). Meet2Learn overcomes these challenges with rule-based inferences and bootstrapping in the case of a new community. The required profile information is used in conjunction with a knowledge source (LSF data for the course schedules) and rule set as an output of association rule learning to infer content-based recommendations.

Collaborative filtering deals in its standard method with a single relation. In Meet2Learn, several relations are encoded into the user profile. From a calculatory perspective, all relations in the algorithm are flattened into a single matrix encoding one subsuming relation. The current system uses standard methods for Collaborative filtering and does not distinguish the different relations. As an extension for next iterations, we will explore alternative approaches for better recommendation results. Some efforts have been put into the development of techniques for dealing with multi-relational data. Szwabe et al. (2013) proposed bi-relational Collaborative Filtering on the basis of RDF-like behavioral data triples that outperforms regular CF. Instead of a user-item matrix, an element-fact matrix is build where the columns consist of the data triples. Anyway, the proposed approach does not support multi-relational data.

Helou et al. (2010) elaborated a multi-relational graph-based recommendation approach, called “3A recommender” that also outperforms standard collaborative filtering. Therefore a multi-relational graph is constructed using the relations of the system entities, namely actors, activities and assets (3A) that are the nodes of the graph. These relations capture interactions between these entities or are constructed through heuristics, e.g. similarity measurements according to latent semantic analysis. An algorithm derived from personalized PageRank is used to generate a ranking vector. This ranking is affected by relative importance of entities, which is manifested as an aspect of contextualization, assuming that entities in the target user’s context are more relevant. This context consists of past activities and the relations that are established by the user, inclusive social relations as “friendship”. In Meet2Learn this might be the user’s interaction with learning groups, extracted relations from external data sources as social networks or other activities that are close in the sense of time.

While these methods solve the problem of multi-relational data, still some flaws and challenges remain. One issue being specific for many domains is the influence of time. The volatile nature of some relations, for example communication between users, is not negligible. Learning groups, the latest activities and interactions that are closer to the present, especially on-going groups, are more relevant to a user. This might have several reasons that can be found in the origin of academic studies, e.g. the diversity of specializations in the field of studies, the failure in exams leading to an anomaly in the course schedule or a reorientation of the student causing not to reflect his former interests anymore. While the overlap of courses of different students is high in the first semesters, it gets relatively low in the later semesters. If time would not be an important aspect, the system will be biased by this initial overlap. Several strategies have been developed to handle time in graph-based systems. In respect to the graph-perspective, collaborative filtering recommendation can be modeled as a link prediction problem (Z. H. Z. Huang, Li, & Chen, 2005). (Z. Huang & Lin, 2009) shows approaches that reformulate the link prediction problem taking time into account, especially considering temporal evolutions of link occurrences. For each time window a single graph is built from all nodes and edges that could be observed in the given time, which leads to successive time slices. A (hybrid) solution is the aggregation of the time series of adjacency matrices into a single one as a weighted summation, where this weighting could be described through different damping functions (Cortes, Pregibon, & Volinsky, 2005).

Taken as a whole, these proposed approaches aim in capturing multiple relations and especially discriminate them in the algorithms – as opposed to aggregating them linearly in a single adjacency matrix. These relations might directly reflect a user’s interest but also ease up the discovery of new items through social network based approaches that are not represented in standard collaborative filtering methods. The specific domain of the Meet2Learn system advises the integration of more extensive content based filtering, such as scheduling and organizational constraints and inferences from static knowledge sources (e.g. LSF), aiming to increase the recall and precision of recommendations.

4. Evaluation

We have conducted a pilot usability and acceptance study with a functional prototype of the Meet2Learn system. The prototype has been evaluated with ten subjects (N=10), according to

Nielson's usability heuristics (Nielsen, 1990). The subjects were asked to follow a scenario in which they played the role of a freshman at the University of Duisburg-Essen in the bachelor program on interactive media and applied cognitive science. The script contained typical user actions with the Meet2Learn scenario, e.g., creating an account, setting up the profile, searching for learning groups, creating learning groups, etc. To overcome the cold start problem for the recommendation system, we created a few hundred "artificial" user profiles to simulate a system state with students in six semesters and multiple learning groups. We observed and supported the users during the run and also asked for instant feedback. Afterwards the subjects were asked to fill out a questionnaire. This questionnaire was based on the standardize Technology Acceptance Model Version 2 questionnaire and has been extended by two open questions (requests for improvements and perceived usability issues) as well as socio demographic questions (age, gender, course of studies, Android experiences). We have used a 7-point Likert scale with the lowest score 1 and the highest score 7.

The first results show an overall positive attitude of the subjects towards the application. In summary, the subjects agree that the application is usable ($M=5.1$) and that they did not perceive the usage as exhausting. Furthermore they didn't find the application hard to explain ($M=4.7$). The importance and the relevance for university studies have been also judged positively ($M=4.7$). From the results of the two open questions and the comments during the runs, we can also conclude and extract valuable information. Subjects suggested including a timetable for organizing the lectures and learning groups in one application. They also would like to extend the functionality of profiles with user pictures and friends lists (i.e., towards a social community). Requests for previewing the members of a learning group before joining confirm the request for more social aspects. One user also requested an alerting component that will notify the user before joining learning groups that take place at the same time as already joined learning groups or lectures marked in the user profile. We also had to face some issues during the run. Although we put a strong emphasis on electing Android skilled subjects, we have observed problems in using the Android platform itself, e.g., software vs. hardware buttons, software keyboard usage etc. Few users had also problems with the creation of learning groups, which we afterwards identified as a software failure on the server side.

Nevertheless, the pilot study shows positive results concerning the usability of the graphical user interface and the acceptance of the approach. We will elaborate on the results of the study and the user comments to improve the usability of the software. Finally, some subjects already asked for the final availability of the software, which indicates a real demand for such a system.

5. Conclusion and Outlook

The Meet2Learn application is an example of integrating and managing information from heterogeneous sources to support personalized and group learning. It takes the heterogeneity of the surrounding information infrastructure as a given and aims at user-side integration. It combines general information from the campus environment with personal profiles and location information. It also includes the use of several types of cloud services.

Recommending learning groups in Meet2Learn uses the user-based collaborative filtering approach. Although the pilot study only used artificial data, we could proof the concept by generating meaningful recommendations to the subjects. Based on the literature review of recommender systems, we will compare different approaches in an experiment to gain more functionalities and better recommendations for the learners.

We plan to have the system ready for use in the winter term (starting in October 2013). It will be further evaluated with a group of beginners from Interactive Media and Applied Cognitive Science. From a systems perspective, the application will be enhanced by using ontologies and semantic processing for identifying specific user interest and needs.

Acknowledgements

We want to thank the master level students of the Meet2Learn project for cooperating in this conceptual and technical challenge.

References

- Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When is “nearest neighbor” meaningful? *Database Theory—ICDT’99* (pp. 217–235). Springer.
- Brainstormer. (2013). <http://brainstormer.collide.info> (last visited May 2013)
- Brophy, J. (2004). *Motivating students to learn*. *British Journal of Educational Technology* (Vol. 36, p. 418). Lawrence Erlbaum Associates.
- Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and UserAdapted Interaction*, 12(4), 331–370. doi:10.1023/A:1021240730564
- Cortes, C., Pregibon, D., & Volinsky, C. (2005). Communities of Interest. *Literary and Linguistic Computing*, 20(4), 383–397. doi:10.1093/lc/fqi048
- El Helou, S., Salzmann, C., & Gillet, D. (2010). The 3a personalized, contextual and relation-based recommender system. *Journal of Universal Computer Science*, 16(16), 2179–2195.
- Giemza, A., Verheyen, P., & Hoppe, H. U. (2012). Challenges in Scaling Mobile Learning Applications: The Example of Quizzer. *2012 IEEE Seventh International Conference on Wireless, Mobile and Ubiquitous Technology in Education* (pp. 287–291). Ieee. doi:10.1109/WMUTE.2012.69
- Hanseth, O. (2010). *Industrial Informatics Design, Use and Innovation*. (J. Holmström, M. Wiberg, & A. Lund, Eds.). IGI Global. doi:10.4018/978-1-61520-692-6
- Hoppe, H. U. (1995). The Use of Multiple Student Modeling to Parameterize Group Learning. In J. Greer (Ed.), *Proceedings of AI-ED 95, 7th World Conference on Artificial Intelligence in Education*. Washington, DC.
- Huang, Z. H. Z., Li, X. L. X., & Chen, H. C. H. Link prediction approach to collaborative filtering. (M. Marilino, T. Sumner, & F. M. S. III, Eds.), 7 Proceedings of the 5th ACMIEEECS Joint Conference on Digital Libraries JCDL 05 141 (2005). ACM Press. doi:10.1145/1065385.1065415
- Huang, Z., & Lin, D. K. J. (2009). The Time-Series Link Prediction Problem with Applications in Communication Surveillance. *INFORMS Journal on Computing*, 21(2), 286–303. doi:10.1287/ijoc.1080.0292
- Isotani, S., Inaba, A., Ikeda, M., & Mizoguchi, R. (2009). An ontology engineering approach to the realization of theory-driven group formation. *The International Journal of Computer-Supported Collaborative Learning*, 4(4), 445–478. doi:10.1007/s11412-009-9072-x
- Jansen, M., Rossmanith, P., Uzun, I., & Hoppe, U. Integrating heterogeneous personal devices with public display-based information services. , IEEE International Workshop on Wireless and Mobile Technologies in Education WMTE05 149–153 (2005). Ieee. doi:10.1109/WMTE.2005.37
- Jansen, M., Bollen, L., Baloian, N., & Hoppe, H. U. (2013). Using Cloud Services to Develop Learning Scenarios from a Software Engineering Perspective. *Special issue on Cloud Education Environments at the Journal of Universal Computer Science (JUCS)*.
- Lam, X. N., Vu, T., Le, T. D., & Duong, A. D. (2008). Addressing cold-start problem in recommendation systems. *Proceedings of the 2nd international conference on Ubiquitous information management and communication ICUIMC 08*, 208. doi:10.1145/1352793.1352837
- Largillier, T., & Vassileva, J. (2012). Using collective trust for group formation. *Collaboration and Technology* (pp. 137–144). Springer Berlin Heidelberg.
- Lucke, U. (2011). A pervasive game for freshmen to explore their campus: Requirements and design issues. *IADIS International Conference on Mobile Learning* (pp. 151–158). Spain.
- Maes, P., & others. (1994). Agents that reduce work and information overload. *Communications of the ACM*, 37(7), 30–40.
- Nielsen, J. (1990). Ten Usability Heuristics. (J. Nielsen, Ed.) *Communications of the ACM*, 3(1990), 1–2.
- Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens : An Open Architecture for Collaborative Filtering of Netnews. In R. Furuta & C. M. Neuwirth (Eds.), *Proceedings of the 1994 ACM conference on Computer supported cooperative work* (Vol. pp, pp. 175–186). ACM. doi:10.1145/192844.192905
- Ryan, R., & Deci, E. (2000). Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. *Contemporary Educational Psychology*, 25(1), 54–67. doi:10.1006/ceps.1999.1020
- Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. (M. Beaulieu, R. Baeza-Yates, S. H. Myaeng, & K. Jarvelin, Eds.) *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval SIGIR 02*, 46(Sigir), 253. doi:10.1145/564376.564421
- Szwabe, A., Misiorek, P., Ciesielczyk, M., & Jedrzejek, C. (2013). Collaborative Filtering Based on Bi-Relational Data Representation. *Foundations of Computing and Decision Sciences*, 38(1), 67–83.
- Weinbrenner, S. (2012). *SQLSpaces – A Platform for Flexible Language-Heterogeneous Multi-Agent Systems* (p. 190). München : Dr. Hut.