Journal of Universal Computer Science, vol. 17, no. 3 (2011), 451-471
submitted: 30/10/10, accepted: 28/1/11, appeared: 1/2/11 © J.UCS

Early Results of Experiments with Responsive Open Learning Environments

Martin Friedrich, Martin Wolpers
(Fraunhofer FIT, Sankt Augustin, Germany,
{martin.friedrich, martin.wolpers}@fit.fraunhofer.de)

Ruimin Shen, Carsten Ullrich
(Shanghai Jiao Tong University, Shanghai, China
{rmshen, ullrich_c}@sjtu.edu.cn)

Ralf Klamma, Dominik Renzel
(RWTH Aachen, Aachen, Germany,
{klamma, renzel}@dbis.rwth-aachen.de)

Anja Richert, Bodo von der Heiden
(RWTH Aachen, Aachen, Germany,
{richert, vonderheiden}@zlw-ima.rwth-aachen.de)

Abstract: Responsive open learning environments (ROLEs) are the next generation of personal learning environments (PLEs). While PLEs rely on the simple aggregation of existing content and services mainly using Web 2.0 technologies, ROLEs are transforming lifelong learning by introducing a new infrastructure on a global scale while dealing with existing learning management systems, institutions, and technologies. The requirements engineering process in highly populated test-beds is as important as the technology development. In this paper, we will describe first experiences deploying ROLEs at two higher learning institutions in very different cultural settings. The Shanghai Jiao Tong University in China and at the "Center for Learning and Knowledge Management and Department of Information Management in Mechanical Engineering" (ZLW/IMA) at RWTH Aachen University have exposed ROLEs to theirs students in already established courses. The results demonstrated to readiness of the technology for large-scale trials and the benefits for the students leading to new insights in the design of ROLEs also for more informal learning situations.

Keywords: Personalized learning environment, Responsive open learning environment, language learning, inter-widget communication
Categories: M.5, L.2, L.3, D.2

1 Introduction

Responsive open learning environments (ROLEs) are characterized through their openness for new configurations, contents and users, and through their responsiveness to learners' activities in respect to learning goals. Openness specifically encompasses the ability to include new learning material and services. These can be combined either in a static fashion or dynamically, therefore allowing learners to create their own learning environments. Consequently, throughout the lifetime of a ROLE, new

configurations will be created by a learner, usually adapted to her needs, requirements and ideas. In addition, ROLEs respond to the actions the learners carry out within them. Such actions can be as simple as downloading a suggested reading and as complex as participating and interacting with other learners in a complete course about a certain topic. Responsiveness then requires the system to respond to these activities, for example by incorporating previously downloaded documents into the suggestions for new readings (as kind of a recommender system) or by offering new services that better suits the needs of learners. For example, a learner might be suggested to use a simulation and conceptual mapping tool to understand a certain problem instead of reading theoretical literature as was provided before.

In the context of the research project ROLE (Responsive Open Learning Environments), a technological-pedagogical infrastructure is developed to enable the creation of individual open responsive learning environments. The ROLE project aims to enable learners to assemble and re-assemble their own learning environments which become advanced *Personal Learning Environments* (PLE) in due course. To get a most comprehensive analysis of requirements for ROLEs, the project is especially targeting highly populated test-beds in different typical lifelong learning settings. Of particular relevance are the critical transition phases of lifelong learning, e.g. from university to company, or from one company to another, when inhomogeneous groups of learners are treated in a one-size-fits-all manner, since there is no way to respond to their individual strengths and weaknesses. Even worse, in such transition phases learners are typically required to become accustomed to working with entirely new learning environments. The main challenge in the current phase is to learn from the many different attempts to let learners create their own learning environments in different learning situations. Typically, higher education institutions are early adopters of new learning technologies, even they are not concentrating on mass individualization of learning as ROLE does. While taking into account the limitations of the presented scenarios we were able to demonstrate two main findings in this paper. The techno-pedagogical infrastructure is ready to serve highly populated learning test-beds with psycho-pedagogical meaningful learning widget bundles. This is a basic requirement to distinguish ROLE approaches from simple Web 2.0 content aggregation environments calling themselves PLEs. The second finding demonstrated the usefulness of the psycho-pedagogical infrastructure in different cultural settings. The ROLE project offers the unique opportunity to try out sets of learning bundles in similar learning scenario, but with a different population of learners. Some early experiences about German and Chinese student learners are presented as well. Although, the main focus of the paper is the technological evaluation.

This paper provides an overview about the general technical concepts used in ROLE in Section 2. A description of the test-beds including the applied scenario, the technical realization and outcomes is done in Section 3, while Section 4 describes first experiences using ROLE in specific test-beds. The results and insights gained from the application of the prototypes in the test-beds are summarized in Section 5.

2 General Technical Concepts

To visualize the potential of ROLE and to gather experience in using the technologies required within the project we are continuously developing and refining a comprehensive technical infrastructure that in its final state will enable the assembly of arbitrary responsive open PLE. In that sense we will not deliver the one and only ROLE PLE, but rather facilitate the emergence of a plethora of PLE assembled by different learners according to their preferences. In this section we present some of the general technical foundations of our infrastructure and its services, before we show their particular application in two different test-bed scenarios and corresponding PLE in Section 3.

One of our most critical overall requirements is to employ open standard Web technologies, in particular an approach based on emergent Web widget technologies such as [OpenSocial, 10; Cáceres, 10]. With an already existing huge amount of such widgets available on the Web, learners can use any widget-enabled public container (e.g. iGoogle, netvibes, social networking platforms such as Facebook, etc.) to assemble their own personal learning environments as combinations of widgets suitable for their individual learning tasks. Such an approach fosters heavy reuse of existing widgets and makes the development of new learning widgets independent from specific learning platforms, thus providing an easy way to extend these platforms with new features by simply adding new widgets from multiple sources. For development purposes we resorted to an early version of Apache Shindig (http://shindig.apache.org), the reference implementation of an OpenSocial container. For the widgets and PLE we implemented in the context of the PLE presented in this paper, we employed an early version of the OpenSocial specification. In Section 4 we will discuss some challenges with respect to this fast moving technology.

One shortcoming of current widget-based PLE is the lack of interoperability between different widgets in terms of communication between them. The transfer of data from one widget to another in the same PLE instance or even across remote PLE instances widens PLE to not just be plain collections of independent widgets. They can rather interact with each other and become combinable to integrated Web applications while following the well-known principles of low coupling accompanied with high cohesion [Stevens, 74]. In our work we employ the concept of *Inter-widget Communication (IWC)* for these purposes. In order to realize an approach with a minimal burden of PLE configuration complexity for learners, it has to be cleanly designed and implemented by developers. As a conceptual approach to IWC the *publish/subscribe* (pubsub) pattern [Birman, 87] is well-suited. For the development of the PLE presented in Section 3 we employed a Gadget pubsub channel [OpenSocial, 10] as a message channel enabling widgets to publish or subscribe information payloads about inter-widget communication events encoded in a unified and extensible JSON message format [Isaksson, 10]. An example of such a payload is provided in Listing 1 for the *selection* of a certain learning object as occurring in the PLE supporting a programming course as presented in 3.2.

```
{type: "namespaced-properties",
event: "select",
userid: "4711",
message: {"http://purl.org/dc/terms/title" = "UML Diagram",
          "http://purl.org/dc/terms/type" = "object",
          "http://purl.org/dc/terms/URI" = http://.../obj-uml.php}
}
```

Listing 1: Example of an Inter-widget Communication Event Payload

Each time a user action is performed inside a widget a respective event is published by broadcasting a message containing information about the user action. In our approach any such message is received by all other widgets subscribed to the same channel. The subscribing widgets then decide whether and how they react to the event depending on message type, content, etc. Naturally, the respective data to be published as well as the behavior on receiving IWC events has to be defined by the respective widget developer. At the current stage, IWC is limited to widgets in the same local browser instance. However, current development efforts lead in the direction of realizing arbitrary forms of real-time communication between learners, remote inter-widget communication, interoperable data exchange, event broadcasting, etc., by employing the *Extensible Messaging and Presence Protocol (XMPP)* [Saint-Andre, 04a, Saint-Andre, 04b, Saint-Andre, 09]. The combination of Web widget and XMPP technologies is feasible, but still challenging, as we will discuss in more detail in Section 4.

For further responsiveness of the learning environment we aim to provide self-evaluation and recommendation mechanisms. Therefore we developed a widget to monitor learner behavior within PLE. Each user interaction inside a widget triggering a publish event is intercepted and recorded by this monitoring widget to observe the learners behavior over time. For this purpose, each received message is transformed into Contextualized Attention Metadata (CAM) [Wolpers, 07] and stored in a central database. Such information recorded for multiple learners enables the generation of recommendations based on collaborative filtering algorithms [Linden, 03] and the provision of self evaluation mechanisms. For recommendations e.g. the current usage history of the learner can be taken into account and compared with the usage histories of other learners and recommend documents of users having a similar usage history as the learner. To provide self evaluation functionality, CAM can be used to visualize and summarize learner activity statistics over time and compare own activities with those of other learners. Furthermore, CAM information can be enriched with manual annotations such as an evaluation of learning success which makes it possible to track the learning progress similar to [Wolpers, 09].

In the following Section we will now demonstrate how the ROLE technical infrastructure and the features described above have been used to support different learner communities and their needs in two of our ROLE test-beds.

3 Two Culturally Different Test-beds

ROLE is based on the vision that responsive open learning environments will accompany learners throughout their learning career, from formal education to learning at the workplace. Several test-beds enable the ROLE project to collect requirements in various settings. In the following sections, we present two culturally different test-beds and the application of ROLE technologies to support their learning tasks including first evaluation results. The first test-bed we describe is operated in the *School of Continuing Education (SOCE)* at Shanghai Jiao Tong University, China, while the second test-bed is operated at RWTH Aachen University, Germany.

3.1 The Shanghai Jiao Tong University Test-bed

SOCE is the online branch of SJTU. Online colleges/universities such as SOCE play a particular role in the Chinese education system. Foreseeing the enormous demand for higher education, the Chinese government decided in 1998 to establish a number of online institutions that were open to those students who did not pass the university entrance exams (in 2007, 10.1 million students applied for 5.67 million places) [People's Daily Online, 2007]. The ROLE SOCE test-bed enables us to learn about PLE from the viewpoints of "average" users, that is learners who are not highly technically literate or who have limited time due to jobs and families.

3.1.1 Test-bed Description

The students on the SOCE (the situation is similar at the other online institutions) are mostly adult learners who have a job. They study two to three years for associate or bachelor degree courses. The courses are similar to regular university courses with the difference that teaching takes place in the evenings and weekends. Each class lasts for three hours. Our blended classrooms are based on the Standard Natural Classroom model [Shen, 08], in which students can either attend the classes in person or attend via the live Web broadcast. Lectures can also be downloaded for asynchronous watching. During the lectures, online students can communicate with the teacher via short messages. The SOCE software system records, encodes and broadcasts each lecture in real-time. Target devices/destinations include desktop/laptop PC via ADSL broadband connections, IPTV devices via the Shanghai Educational IPTV channel, universities in western China via two satellite connections and mobile devices via GPRS. SOCE produces about 6GB of content every day. The current SOCE learning management system supports student-student communication via forums. Teachers can create exercises/homework to be completed by the students.

Some of the gaps in the current learning processes are representative for most Chinese educational settings. Students are degree/certificate oriented: the primary goal is to receive a formal acknowledgment rather than the mastery of the subject itself. Group work is hard to perform as it is seldom performed at school. In consequence, the practical application of learned knowledge is difficult for Chinese students. For instance, in language learning, the students have extreme difficulties communicating with native speakers. They are insecure, shy, and make frequent mistakes. Furthermore, in our lectures we over and over make the experience that the students do not know, have difficulties or simply do not use existing tools such as

online dictionaries, pronunciation tools or micro-blogging sites. Reasons given include lack of time, interest and motivation. In ROLE, we experiment with different scenarios to address these problems.

We experimented with an Open Learning Environment and a variety of tools (video conferencing, micro-blogging, translation services, text-to-speech, etc.) over two semesters in the courses English Listening and Speaking, German I/II, French I/II and Introduction to Computer Science. About 50-100 students attended the language learning courses, and about 1.200 students took part in the Computer Science class. From our experience we knew that the students at SOCE have limited knowledge about Web tools (RSS is virtually unknown), only limited time at their disposal and limited technical expertise. Furthermore, in the Confucian culture of China learning is still very teacher-centered [Zhang, 07], and students are not used to actively contribute within a class. We therefore decided to build PLE according to the suggestions of the teachers and make these pre-built PLE accessible to the students. The teachers presented the PLE in class and showed example usages. In order to encourage usage, the students were assigned homework that required using those PLE. We then observed the students' usage of the PLE and collected feedback from the teachers and students by interviews and questionnaires.

3.1.2 Scenario

For testing the ROLE technologies within the SJTU test-bed we defined the following scenario. The French teacher assembled the PLE shown in Figure 1. The widgets on the left hand site consist of tools helpful in a language learning context, namely a translator widget, a spell checker, a text-to-speech engine and a recording widget. In a first experiment, the PLE was introduced to the students during class and their usage was illustrated by examples. Then, the students received the homework to author and record a self-introduction using the specified tools. The intention was that the students, which were beginners in French with almost none French knowledge, would use the translation tool to translate English or Chinese source content into French, fine tune the translation with the help of the spell checker, listen to the pronunciation of the text using the text-to-speech widget, and finally record their own pronunciation using the recording widget an compare it with the given pronunciation of the text-to-speech widget. The outcome of this scenario which was used in a class is reported in 3.1.4.

3.1.3 Specific Technical Realization

For the experiments at the SJTU test-bed, we applied a version which is based on the portal Liferay. While we did not offer advanced features such as inter-widget communication, the basic functionality was the same and enabled us to collect information about the usage of an OLE in a higher education context.

Liferay is an open-source portal system that enables the creation of groups, group-specific pages, and the population of pages with widgets. In the SJTU setting we created groups for each class (French I, II, computer science) as well as user accounts for each student in class and for each teacher. The teachers then added a number of pages for each of their lectures. For instance for French I, the pages “Tools” (a number of tools helpful for translation and speaking practice tasks) and

"Grammar" (a grammar book that was collaboratively created during class) were created and extended with a large number of exercise pages that contained multiple-choice exercises for testing several aspects of the French language. Figure 1 shows an early version of this system.

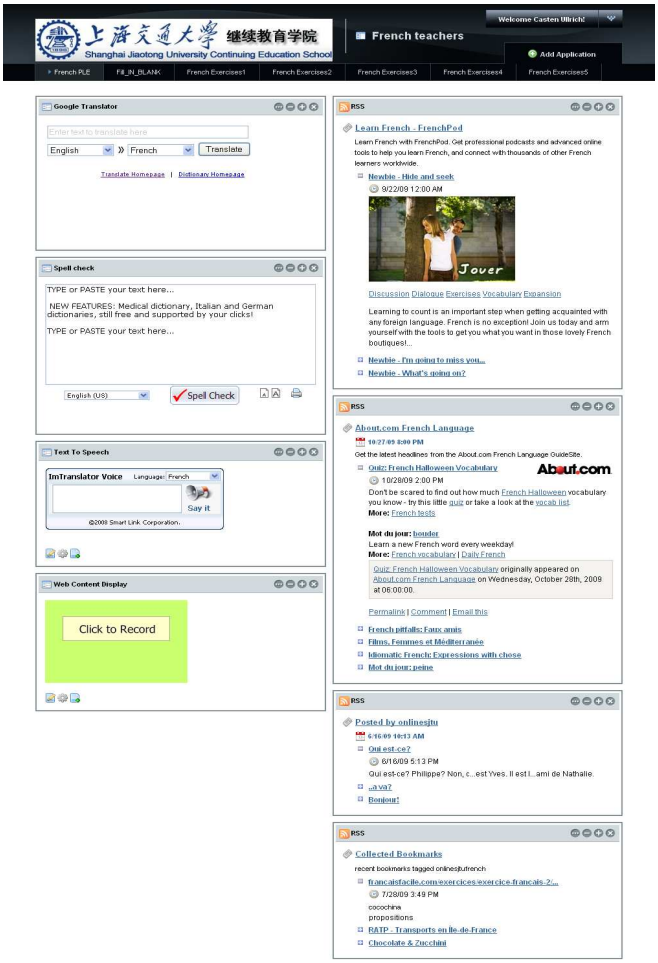

Figure 1: Screenshot of the SJTU prototype

The included widgets were embedded in two different ways: either according to the Java Portlet Specification (JSR-168) or they were embedded using a Liferay widget which allowed the inclusion of arbitrary HTML code. Widgets which are included using the JSR-168 specification are the RSS widget and the multiple-choice exercise tool. Most of the language related widgets are included using the HTML method, namely the spell-check widget, the text-to-speech widget, the recording widget and the etherpad widget which is used to simulate a joint grammar book. The current version of Liferay supports OpenSocial which enables a more general

inclusion of widgets. This functionality was not completely available during the reported period; using a prototypical version we included a translation widget.

3.1.4 Evaluation

Here, we will briefly report on our experiences with using the PLE in the SJTU test-bed for language learning support, namely the classes French I and French II over the three semesters Spring 09 (S1, June 2009–August 2009), Summer 09 (S2, September–November 2009), and Spring 10 (S3, starting March–June 2010). The classes were taught in French and English. Numbers about students inscribed for the French lectures are difficult to give. The school administration has figures about the total numbers of students in each tier, but no information about the selected second language as this can be freely chosen (students have the choice between German, French and Japanese). The S1 tier (in which French I was taught) counted 128 students, S2 474 students (347 French I tier, 130 French II tier) and S3 369 students (178 French I, 191 French II). The students progress from French I to French II. This means that the S2 French II students are mostly identical to the S1 French I students, etc. Similarly, concrete numbers about attendance are not available: students do not attend regularly due to their busy schedule. They can come to class in person, watch via their browser live or later download the video of each class. In average, about 20–30 students attended the French I lectures in the classroom and about 10–15 students the French II lectures. An indication of participation is given by the numbers of students who took the final exams: 33 for S1 (French I), 106 (French I: 72, French II: 34) for S2, 95 (French I: 37, French II: 56) for S3. This is about 23% of the total numbers in each tier. These numbers are typical for the second language (which is deemed as rather unimportant by the students). The lecturer was popular among the students and received good grates in students' evaluations.

We employed several approaches to capture qualitative and quantitative data. The research team observed teacher-student interactions during the lectures to learn about the general interactions. Additional information was obtained by collecting feedback from the teacher. A questionnaire and a video-taped Q&A session served to collect information about the impressions of the students. Furthermore, we analyzed the log-files of the PLE server in order to gain quantitative data about students' usage of the PLE. While the PLE was introduced and used in class, the students were expected and encouraged to use it outside of the class. The student accounts were automatically created using the list of students inscribed in each tier. An analysis of the PLE logging data showed that for S1 and S2, 49 students logged in at least once which is about half of the 106 students who took the exam in S2 (recall that this number includes the S1 students). For S3, 38 students logged in at least once. These numbers are comparable to the amount of students who handed in homework.

We believe that this rather low uptake rate of the PLE was mainly due to how the PLE was introduced in class. For the students it is not sufficient to present a single example of how to use the OLE in class. In itself, this will not enable and motivate students sufficiently to work with the provided tools. We hope to improve these figures by increasing the incentive to use the PLE. Measures will include extra points for the final grade but also better communication of the value of a PLE. In particular, students need to understand how the tasks and services work which can help them to achieve their goals. Each OLE usage needs to be broken down into individual steps.

For instance the task of doing a spoken self-introduction can involve the steps of writing the introduction in the native tongue, translating it, polishing it, using a text-to-speech tool to listen to it, a recording to practice one's own pronunciation and finally recording and publishing it. Demonstrating this whole sequence only once or twice overtaxes the student. Each single step needs to be shown and done by the students several times. The single steps as well as the combination of services should be assigned as homework giving the students an initiative for practicing. Breaking down the usage of an OLE helps students to understand how to use it. Even more important is that the students understand why they should use a PLE. Students need to see the value of performing additional tasks which are not directly related to language learning.

One particular feature frequently mentioned in favour of PLE is that users (students and teachers) can configure and assemble them depending on their needs and with tools they are already using. We investigated whether our students actually have a need for these features and what kind of tools they are using. The data was collected using a Question and Answer session which was videotaped and by a questionnaire. 65 students completed the questionnaire which was completely voluntary. According to the data, our students are only using few of the available tools. 20% use blogs, 6% forums, 15% social networks, 6% RSS, and 85% chats. None of the students uses Wikipedia (which is not blocked in China) and only 3% use translation tools. 63% do not access the Internet with their mobile phone.

For western students the access to the internet is almost as common as watching TV. Applying a web based PLE to other countries requires taking into account country-specific restrictions. For example as SJTU is located in China the access to quite a large number of Web sites including social networks such as Twitter, Facebook and Friendfeed as well as many Web services is blocked. Furthermore, many institutions such as schools or companies restrict the access of sites deemed to be inappropriate for their scholars.

Furthermore, we experienced new directions for technical research which should be taken into account for the development of the next ROLE prototypes. This includes the necessity to create accounts for some services which require a login or the necessity to have a single sign on feature since logging in several times leads to frustration of the students. Finally, even though logging data was collected in the SJTU test-bed, students did not raise privacy concerns. On the contrary, repeatedly, students uttered concerns that their contributions in the employed tools might not be noticed by the teacher.

3.2 The RWTH Aachen Test-bed

Another test-bed is provided by the "Center for Learning and Knowledge Management and Department of Information Management in Mechanical Engineering" (ZLW/IMA) at RWTH Aachen University, Germany. The ZLW/IMA is responsible for the lecture "Informatics in Mechanical Engineering" with more than 1000 students each time. The usage of e-learning tools supports the teaching. One of these e-learning tools is the *Web 2.0 Knowledge Map (WKM)*, the RWTH Aachen test-bed. The WKM is a semantic-net based tool which is similar to a Wiki and its supports a non-linear learning approach.

3.2.1 Test-bed Description

Our course "Informatics in Mechanical Engineering" is held in the second term of the study path for the Bachelor in mechanical engineering and in the fourth term in study path of industrial engineering. It is composed out of three different course formats: The theoretical foundations are taught in a lecture. In the project task, the second one, students train practical foundations. In a third part two weeks before the final exam, the students are prepared for the exam in small sized group-courses. During the project "WeKnow" (finished in 2007), the project task was chosen as the optimal course format to implement the WKM. In the project task, the students use a self-directed learning approach [Sattari, 06]. The students work in groups of five to six persons and have to implement a tool to simulate a work process at a terminal station [Henning, 03]. The project task takes place in a computer pool with 220 workstations. Within this project it is intended to get an overview over the usage of the WKM for a longer period and over several generations of students.

The WKM provides selected contents of the course to the students. Subject-specific content is linked and offered by means of various elements such as images, audio and video files, animations or podcasts. The WKM helps the students to solve individual subtasks they should solve during the project task. The goal of the WKM tool is to guide and to support the self-learning process of the students. The tasks of the teachers focus on motivation, introduction and provision of high quality basic knowledge.

An analysis of the number of users of the WKM during the WeKnow-Project indicates that the students mostly use the WKM during the term. However, two weeks before the exam (in parallel to the small sized group-courses) the WKM usage increases enormously as the tool is used to support the students in their exam preparation.

The original WKM was integrated into a prototype described in the next section, which was developed before the summer term 2010. During the summer term, the usage of the WKM was tracked with the CAM-widget which was developed in context of the ROLE project. In the last week of the term, a survey was carried out. First results of an evaluation are presented in 3.2.4.

3.2.2 Scenario

In the RWTH Aachen University test-bed scenario we assume on the one hand that a learner wants to improve his knowledge about informatics in mechanical engineering and on the other hand a teacher (for the lecture "Computer Science in Mechanical Engineering", a 2nd term lecture) wants to provide a learning environment for his students as an additional support during the learning process. This tool should be used in parallel with traditional teaching methods such as lectures and exercises.

In the second part of the scenario a teacher chooses widgets from a widget store and arranges them to an integrative learning environment regarding to the content of the lecture "Computer Science in Mechanical Engineering". In our case, for students which prefer a non-linear learning approach this tool is a good choice because it provides many advantages. Particularly for students - we call them digital natives, which are familiar with remote controls, PCs and cell phones - the learning environment is intuitively comprehensible. Their specific knowledge and skills such

as scanning skills, multi-tasking skills and the ability to process discontinued information have to be addressed [Sattari, 06]. Thus, the *Knowledge Map-Widget* is chosen as the main part of the learning environment. Furthermore a *chat widget* would support the collaboration and a *history widget* for the last five visited items helps the students to maintain the overview in a non-structured environment. As a result the widgets are arranged as shown in Figure 2. The link to the learning environment, labeled WKM, is named in the lecture and on the homepage of the institute. The teacher will test this combination of widgets and will update the setup before each term, if she recognises (by evaluation) that the students want to use another setting.

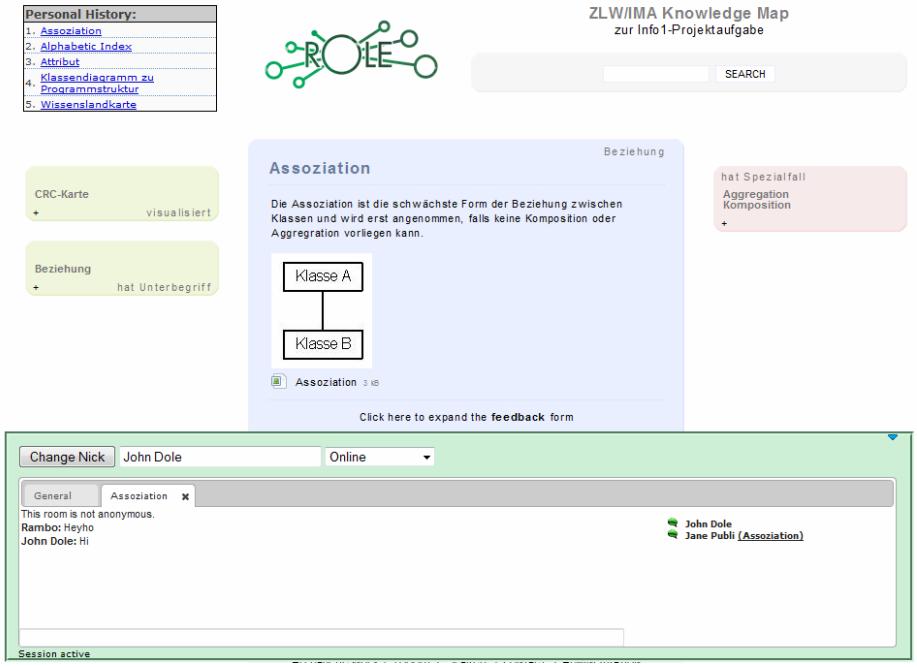

Figure 2: Set of the Web 2.0 knowledge map in summer term 2010

In another scenario, a 2nd term mechanical engineering student receives the link to the WKM in the lecture. In the tutorial "project task", which takes place parallel to the lecture, the student gets different practical tasks to solve, e.g. the creation of a "UML class diagram". The student remembers the tool, searches and finds the word "class diagram". Now the student can only use the original *WKM-widget*. In this case she only needs to read the description of the selected item and receives hints for more relevant items at the left and right side of the description. By using the *history-widget,* the student can track back the last five selected items. Additionally, the student can discuss the task and the class diagrams via the *chat* function. Whenever the student has to solve a task or does not know a new item in the lecture she can use the knowledge map as a reference book and as a tool for collaboration.

3.2.3 Specific Technical Realization

In this section we briefly discuss the key points of the technical realization of the RWTH test-bed. In particular, we discuss how we integrated the original WKM [Hees, 06] into the ROLE infrastructure including Inter-widget Communication and Monitoring. In particular, we created a widget-mashup with enhanced WKM features such as the learning history and multi-user chat widgets described in the previous section.

The technical foundation for the ROLE-enhanced version of the WKM is Apache Shindig (http://shindig.apache.org/), the Open Source reference implementation of an OpenSocial [OpenSocial, 10] container. Here, we mainly used the container functionality to setup the WKM as widget-based PLE. For the WKM, there was no requirement of letting students choose from a catalogue of widgets, so the container itself consisted of a simple HTML file including a static layout for three different widgets, which we will describe in the following. Although all of these widgets were compliant with the OpenSocial specifications, their purposes, used technology and development processes were completely different, demonstrating the importance of an infrastructure that allows the integration of highly heterogenous technologies at the time of PLE usage. Given the Inter-widget Communication infrastructure, see Section 2, the integration of all three widgets into the test-bed PLE could be achieved with a comparably low effort. Central element of the resulting PLE is the WKM Widget, a "widgetized" version of the original WKM. Using the OpenSocial Gadget specification, such a widget could theoretically be constructed easily by just providing the URL of the original WKM in order to wrap it into a widget. However, any modifications to the original within the widget code are not possible anymore with the regular OpenSocial Gadget API. Thus, we decided to utilize the PMRPC framework (http://code.google.com/p/pmrpc/) to communicate between the original WKM-URL and an iFrame in the Widget.

In order to foster direct communication among the test-bed students in their course and to push forward the envisioned ROLE real-time communication and collaboration infrastructure, the multi-user chat widget was developed using a combination of open standard OpenSocial/AJAX and XMPP technologies [Saint-Andre, 04a; Saint-Andre, 04b; Saint-Andre, 09; Saint-Andre, 10]. The main purpose of this widget was to provide students with a communication medium for discussions on the topic of software development in the context of the project task in general or on a specific topic offered by the WKM. Therefore, the widget supports multiple tabs, each representing a specific chat room. The additional support for handling real-time information on presence and status additionally offers students a certain level of community awareness. In our case we used this concept to indicate on which topics students were currently active in order to bring them together based on their current interests.

The main purpose of the history widget is to record Contextualized Attention Metadata [Wolpers, 07] by intercepting all events exchanged via the local Inter-widget Communication mechanism we employed in this test-bed. However, since this functionality by itself will not bring any direct benefits for the students, we decided to not only record, but also to show the last recorded events in the form of a history list of the last visited WKM topics as a tool for students to track their recent learning activities. Technically, the CAM widget intercepts all Inter-widget Communication

events and then interacts with a Web service responsible for persisting and retrieving CAM data.

As previously indicated, all three widgets do not only provide their functionality in a stand-alone fashion, but also interact with each other, thereby realizing the WKM as a complete integrated Web application, but with the usual benefits of decoupling during the development process. In that sense, the majority of widget functionality could be developed independently, and the integration worked seamlessly by simply exposing the respective Inter-widget Communication events and their data representations as interface definitions. The following three examples of such events demonstrate some of the interactions we implemented:

- *Entering a Topic-based Chat Room on Topic Selection:* when a user selects a topic from the WKM, a corresponding chat room is entered in the chat widget. At the same time his online status is changed to the new topic in real-time and visible to and clickable for other students.
- *Following a User's Activity:* when a student clicks the online status of another student, he navigates to the corresponding topic in the WKM, in turn triggering an event to enter the corresponding topic-specific chat room.
- *Recording Inter-widget Communication Events:* all events published by any of the other widgets are intercepted and processed by the CAM widget. CAM data is stored, and the history list is updated by the underlying Web service. A click on any list item triggers selection of the corresponding WKM topic, the corresponding chat room is entered and the online status changed accordingly.

The above examples demonstrate that it is even possible to build complete chains of events. Although such chains might be a powerful concept of creating complex applications, there is a certain risk of building event cycles, in turn resulting in infinite loops and an annoying user experience due to the feeling of having lost control over the interface.

3.2.4 Evaluation

In this section we provide a brief report on our experience with and evaluation of the first version of the "widgetized" version of the WKM, introduced in the lecture "Informatics in Mechanical Engineering" at RWTH Aachen University in summer term 2010 (April 2010 – September 2010). The teaching period took place from Mid-April until end of July. The lecture is taught in German. The lecture is an obligatory lecture in the 2nd semester for students in the field of mechanical engineering and in the 4th semester for students of the field engineering economics. The real number of users is difficult to determine. The potential number of users is about 1400 students, because this is the number of students who attend this lecture in their field of study. In this summer term 1194 students have written the final exam, which indicates that not all students have attended the lecture. The completion of the project task, a weekly 2.5h practice course, is voluntary. The knowledge map is mainly introduced in the project task as one e-learning tool, students can use as an addition to other tools e.g. the lecture-script and a wiki called eClara. Students may also use the knowledge map outside the course.

We used several approaches to capture user data. We used the history-widget (CAM), which collects each interaction with knowledge map and the chat by the students, see 3.2.2. As a second tool we implemented a survey dealing with the opinion of the students about the knowledge map. We also collected feedback from users while they used the knowledge map (there was an integrated feedback form included into the knowledge map).

While the knowledge map (including the Chat- and the History-Widget) was introduced in the lecture and in the project task, the students were encouraged to use it also outside the course. The students get logins to use the workstation in a designated computer pool. They can also use these logins to use the knowledge map. Thus, students have to remember only one login for the workstation, the knowledge map and also for lecture relevant downloads such as slides, old exams, etc..

As shown in Figure 2 here is a minimization-button (arrow symbol) at the top right corner of the chat widget. In April this button was not introduced yet. A first feedback from the students (by using the feedback form) showed that they think that the chat occupies too much space. They wanted to personalize the version of the knowledge map but nevertheless think that the chat is useful, but that it should be closed and indicate a new message by flashing or alternative clues. As a first quick reaction we included the minimize/maximize button.

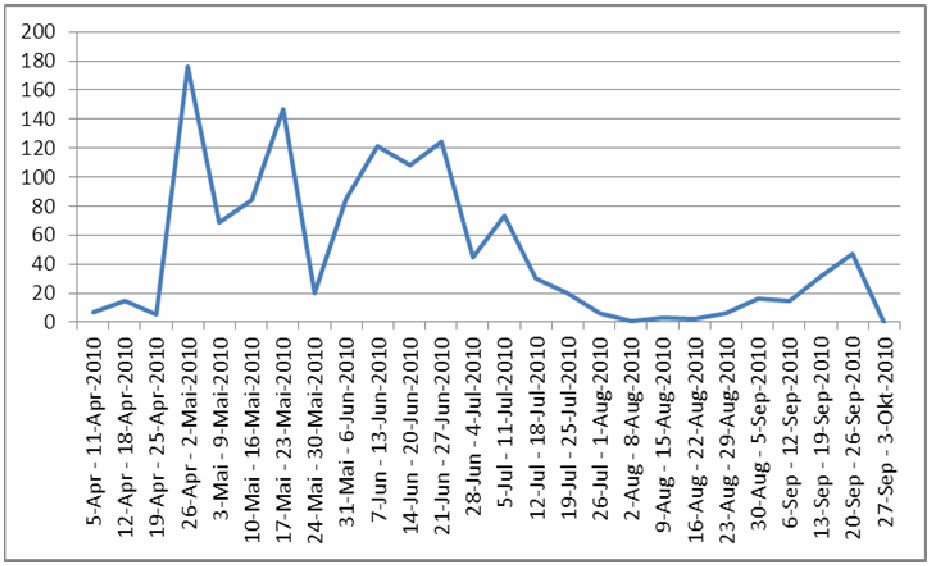

Figure 3: Visitors per week (generated by analysis of CAM-data)

The analysis of the CAM-Data illustrated that about 180 students have tested the knowledge map during the first week of the project task, see figure 3, 26-April – 2-Mai-2010. The next two weeks of the project task include a self-introduction to the programming language C++ by using the tool eClara. Having completed this task the content of the knowledge map is more relevant. The week 24-Mai – 30-Mai-2010 was the so called “excursion-week”, a week without any lectures and exercises. In the last three weeks students implemented a tool for a “goods station”. In this period the

actual content of the knowledge map is not really relevant. The final exam was written on 25-September 2010. The small peak before this date illustrates that some students used the knowledge map outside their courses for the preparation of the exam. Not surprisingly, it is shown that the top selected items and the top search keywords correspond to the content of the lecture and the project task.

Our survey about the usage of the knowledge map showed that many students (about 80%) do not know that there is a tool called knowledge map which can be helpful for an exam preparation. Those who know the knowledge map and used it think that the knowledge map can help them to promote their skills in "Subject-specific content", "dealing with e-learning tools" and "Networking". They also like to work with a knowledge map in other courses. The students also "consider the knowledge map a good idea". A comparison with the survey-results of 2008 indicates that the students in 2010 have less used the knowledge map.

The new integrated chat-function was hardly used by the students. So it is not surprising that the students evaluated the function negatively. The knowledge map was used mostly within the project task. There were also direct contact persons which were used instead of the chat. Furthermore a chat-function is more useful if many users are online at the same time; this was not the case outside the project task.

The data shows that in 2008 and 2010, only few students have used the WKM. Among the students using the WKM the acceptance is high. This suggests that the reason why most students do not use the WKM could be a lack of recognition of the additional value the knowledge map can provide. Hence, the intrinsic motivation to use the WKM is not high enough. For these reasons we consider various solutions for this problem. One solution depicts a better integration of the WKM into the concept of the lecture. By this means the students have to use the WKM initially and thereby identify the benefits more easily. Another possibility is to add more helpful content to the WKM. Such additional content could be different links to You-Tube-videos (played in a You-Tube-Widget developed by ROLE-partners) or animations of pictures to visualise specific processes. Concerning this, it is an advantage that the contents of the course will develop towards "robotics" in the next summer, because a huge variety of films is available in this research area. We believe that these measures will increase the acceptance of the WKM.

4 Discussion of Development Experiences

After a discussion of the usage of the developed PLE by learners, we use this section to also provide a discussion of experiences from the perspective of developers. As we indicated in the previous sections, the ROLE project is targeting to create an infrastructure for learners to create their own PLE from tools already existing. However, since our infrastructure introduces opportunities for interesting new classes of widgets and combinations thereof, we aim to attract Open Source as well as commercial developer communities on a large scale as a long-term goal in order to create a critical mass of learning tools and people that care for self-sustenance even before the project ends. Practically, this also meant for us that we would have to design our infrastructure suitable for distributed widget development processes only by exposing interfaces for communication with other tools, services and widgets. In this section we will discuss some of the experiences we made during the development

processes for the aforementioned test-bed PLE from the developer perspective, thereby demonstrating that distributed independent development and subsequent integration with the ROLE infrastructure is already possible, but yet challenging.

As we aimed to build widget-based PLE, we first scouted for a development environment suitable for hosting and rendering the sets of widgets and services contributed by the distributed development partners. The goal was to organize our process according to the following policy in order to emulate the scenario of distributed widget development and subsequent integration into the ROLE infrastructure by different independent parties:

- Every developing partner maintains a local development environment.
- One partner maintains an additional integration environment, from where all distributed results of the partners should be usable.
- All local development environments and the integration environment are fully compliant with each other.

The second policy intended to emulate the future situation of combining multiple widgets from different sources into own PLE hosted at arbitrary ROLE-compliant containers. Here, we put a strong focus on staying compliant with already established public containers such as iGoogle, netvibes, Liferay, as well as manifold social networking sites such as Facebook that already include widget support for small stand-alone applications such as mini-games.

Especially the mutual compliance required in the last policy rule was not fulfilled for reasons discussed in the following. Although there exist multiple Open Source solutions for widget containers already, the biggest challenge was (and still is) their instability as a result of the fast moving OpenSocial specification process, especially regarding IWC, one of the main requirements for our PLE.

We tested three different development environments, mainly consisting of OpenSocial containers and their surrounding infrastructure, i.e. Web servers, application servers, and other tools supporting the development process.

1. SocialSite (based on Shindig within Glassfish) [SocialSite, 10]
2. Apache Shindig [Shindig, 10]
3. OSDE (Eclipse Plugin with integrated Shindig) [OSDE, 10]

With all of the above systems we encountered at least one of the following problems:

- Lack of Forward Incompatibility
- Client Browser Dependence
- Server Platform Dependence
- Inaccessible Bugs in Generated Code
- Lack of Developer Support
- Incompatibility with External Libraries
- Instability of Draft Standard Specifications
- Missing/Changing Libraries for inter-widget Communication

The first problem was related to container installation, in particular with SocialSite. First, the current SocialSite distribution was restricted to specific, already outdated versions of Glassfish and Shindig, and thus is not forward compatible - an essential property, when working with experimental systems likely to evolve in future. Given the diversity of devices and platforms used by the different developing partners, we quickly had to find out that platform and browser independence was not given at all. Although both Shindig and Glassfish are Java-based, and all partners were using the same installer version, some partner installations worked as expected, while others exhibited unexplainable container side errors. Another issue was browser incompatibility. Even with a fully working container, some browsers did not render widgets correctly. This problem is highly likely to be related to different implementations of JavaScript engines implemented in current browsers. Another essential problem was the inaccessibility of bugs in JavaScript code. In many cases, problems occurred outside the source code under developer control. The reason was a malfunction in the code production performed by the container itself to render the widgets. Furthermore, error messages were often cryptic and incomprehensible and thus did not provide any hint to the original location of an error. In conjunction with the above problems we had to experience that developer support by the SocialSite team was not available at all. At the time of writing this document, it seems quite obvious, that SocialSite is dead. After the initial experiments with SocialSite, we communicated possible alternatives. The alternative usage of Apache Shindig instead of SocialSite was also not successful for all our developers at that point in time. However, due to the recently finished incubation process at Apache, we can expect to receive support and more mature versions from the Shindig team in future. The OpenSocial Development Environment (OSDE), which also uses a built-in Shindig test server, was considered as a helpful tool with respect to the development of single widgets, but proved to be impractical, when it comes to the development of multiple intercommunicating widgets. Further problems were related to the incompatibility of external JavaScript libraries with the Widget container, which resulted in strange code rewriting effects or cross-domain issues. With special regard to the pubsub-based inter-widget communication approach pursuing in ROLE, we experienced that at the current time Shindig is in a transition phase of changing from the pubsub feature included in the Google gadget API to another pubsub mechanism included in Open Ajax Alliance Hub 2.0 [OpenAjaxAlliance, 10]. Regarding all of the above issues, we can state that the collaborative distributed development process of a PLE was challenging due to the lack of a common stable and reliable development environment. However, the situation has by now improved tremendously, since Shindig's incubation process has ended, and the OpenSocial specifications converge more and more to a stable core.

With the long-term goal in mind to create a class of ROLE widgets supporting arbitrary forms of real-time communication between learners, remote inter-widget communication, interoperable data/metadata exchange, event broadcasting, etc., we put a special focus on the *Extensible Messaging and Presence Protocol* (XMPP) [Saint-Andre, 09]. Due to its rich built-in properties and core features, its set of official *XMPP Extension Protocols (XEPs)* and the variety of application use cases in ROLE, we explored XMPP client development support especially in conjunction with

widget technologies and JavaScript libraries. In the following we will discuss our experiences during the development of the PLE from Section 3.

Although there exist quite a number of XMPP libraries for JavaScript, we learned that many of them are still insufficiently mature at least for the realization of our goals, and definitely need improvement. In particular, we experimented with a selection of libraries listed on the official XMPP website (http://xmpp.org). A comparison of these libraries is presented in Table 1 and is thought to support developers in their choice.

Library	Connection Technique	Code/Documentation Maturity	UI Support	Supported Features & XEPs
dojox.xmpp [Dojo, 10]	XMPP over BOSH	ok/weak	yes	Core: IM, Presence, Roster XEP-0004: Data Forms XEP-0030: Service Discovery XEP-0085: Chat State Notifications XEP-0206: XMPP over BOSH
xmpp4js [XMPP4JS, 10]	XMPP over BOSH	ok/ok	no	Core: IM, Presence, Roster XEP-0004: Data Forms XEP-0030: Service Discovery XEP-0049: Private XML Storage XEP-0077: In-Band Registration XEP-0085: Chat State Notifications XEP-0100: Gateway Interaction XEP-0206: XMPP over BOSH
strophe.js [Strophe, 10]	XMPP over BOSH	ok/weak	no	XEP-0206: XMPP over BOSH
js.io [JS.IO, 10]	(CSP) Comet Session Protocol	weak/weak	no	unknown

Table 1: Comparison of JavaScript Library Support for XMPP

Probably the most important notion was that libraries exhibited different levels of code and documentation maturity and different sets of core and extension protocol features implemented. The minimum level of functionality realized by all of the above libraries is the emulation of persistent, stateful, two-way connections to an XMPP server using techniques such as *COMET* or *Bidirectional-streams Over Synchronous HTTP* (BOSH) [Paterson, 08], often in conjunction with additional libraries for XMPP XML stanza construction and parsing. The next level of functionality was the implementation of XMPP core services, i.e. instant messaging, presence, and roster lists, which were only realized by dojox.xmpp and xmpp4js. Functionality beyond XMPP core services often only include very basic XEPs, e.g. for service discovery, server ping, data forms, etc., but does in all cases not include the implementation of central XEPs supporting powerful communication techniques such as multiuser chat, publish/subscribe, etc. Although code and documentation quality as well as the set of supported XEPs in xmpp4j seemed superior, we decided to start the experimental implementation of further XMPP XEPs based on dojox.xmpp, because it provides cross domain solutions, is integrated into the well-established dojo toolkit, and is designed to make use of the dojo widget framework for the provision of custom

XMPP-powered UI elements. By now, we have extended dojox.xmpp with implementations of high-level support for various official XEPs, among them Multi-user Chat [Saint-Andre, 10], PubSub [Millard, 10] and smaller XEPs such as In-Band registration or data forms. As future work, further central XEPs will be implemented and evaluated in a whole class of XMPP-enabled Widgets.

Drawing the conclusions from our experiences, we can state that the technologies we experimented with were insufficiently mature for the deployment of a stable integrated prototype ROLE PLE assembled from a set of innovative tools realized using a combination of different bleeding-edge Web technologies. However, we finally managed to deploy prototypes for all five test-beds in their own productive environments as well as in Graaasp [Bogdanov, 10] in a rather stable version, but still with a lot of open issues to be tackled in later development stages of the ROLE project, especially with regard to providing easy-to-use libraries for external developers to enable large-scale extensions of the universe of ROLE-enabled learning tools.

5 Conclusion and Outlook

In this paper we provide an overview of general technologies and concepts for creating Open and Responsive Learning Environments. We applied these technologies in two different test-beds, both using different scenarios. Results gathered in these contexts show us, that Responsive Open Learning Environments create enormous challenges on the psycho-pedagogical as well as on the technical side. Some of the challenges were sketched in this paper. The ROLE project has chosen a spiral “organic” development and deployment process to cope with the challenges mentioned above. We started with very small scale projects only covering partial requirements of the five test-beds of ROLE. This was necessary to understand the different development cultures among the developers at the different ROLE sites and also the language of the end users stating their requirements - an experience that will surely help for the larger roll-out planned. Indeed, beside those challenges the interplay between end users of personal learning environments and developers of enabling technologies and products is crucial. It is not only a problem that the current generation of developers does not recognize the time horizons covered in the development of personal learning environments, thus making industrial scale standards deployment inevitable, but also open source developers and company developers do not understand each other. At the moment, flexibility and openness is more on the side of the company developers.

In the near future, the development process is widened to cover more test-beds. This is due to the necessary learning experience for developers that results developed in the context of one scenario are not necessarily transferable other scenarios. As the ROLE project is targeting the transitions of learning, the future scenarios involve learners' shifts of interests and learning goals during the scenarios like leaving the university with some degree of self-regulated learning and entering a company where learning goals are only valid in the light of the company's strategy.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no 231396 (ROLE project).

References

[Birman, 87] Birman, K., Joseph, T.: Exploiting virtual synchrony in distributed systems. In *Proceedings of the eleventh ACM Symposium on Operating systems principles (SOSP '87)*, 1987. pp. 123-138.

[Bogdanov, 10] Bogdanov, E., Salzmann, C., El Helou, S., Sire, S., and Gillet, D.: Graaasp: A Web 2.0 Research Platform for Contextual Recommendation with Aggregated Data. In ACM Conference on Human Factors in Computing Systems (HCI) - Work-in-Progress, 2010.

[Cáceres, 10] Cáceres, M.: Widget Packaging and Configuration. W3C Working Draft. October 2010. http://www.w3.org/TR/2010/WD-widgets-20101005/ (last access: October 2010)

[Chakrabarti, 99] Chakrabarti, K., Mehrotra, S.: The Hybrid Tree: An Index Structure for High Dimensional Feature Spaces, In Proc. Int. Conf. on Data Engineering, Feb 1999, pp. 440-447.

[Cocoon, 02] Cocoon XML publishing framework, 2002, http://xml.apache.org/cocoon/ (last access: Oct 2010).

[Dojo, 10] Dojo Toolkit. Unbeatable JavaScript Tools. http://www.dojotoolkit.org/ (last access: Oct 2010).

[Glassfish, 10] Glassfish Open Source Application Server. https://glassfish.dev.java.net/ (last access: Oct 2010).

[Hees, 06] Hees, F., Backhaus, W., Sattari, S., Henning, K.: A web-based knowledge map for higher education. In: Conference Proceedings: 22nd ICDE World Conference. Rio de Janeiro, Brazil, Sept 3-6, 2006.

[Henning, 03] Henning, K., Kutscha, S.: Informatik im Maschinenbau. Aachener Reihe Mensch und Technik, Band 31. Aachen: Wissenschaftsverlag Mainz in Aachen, 2003.

[Isaksson, 10] Isaksson, E., Palmér, M.: Usability and Inter-widget Communication in PLE. In *Proceedings of the 3rd Workshop on Mashup Personal Learning Environments (MUPPLE10), Barcelona, Spain, September 29, 2010,* CEUR Workshop Proceedings Vol. 638, 2010.

[JS.IO, 10] js.io. Real-time Web Applications. http://js.io (last access: Oct 2010).

[Linden, 03] Linden, G., Smith, B., York, J.: Amazon.com Recommendations - Item-to-Item Collaborative Filtering, IEEE Internet Computing 7(1), 2003.

[Millard, 10] Millard, P., Saint-Andre, P., and Meijer, R.: XEP-0060: Publish-Subscribe. Technical report, XMPP Standards Foundation, Draft Standard. July 2010.

[OpenAjaxAlliance, 10] OpenAjax Hub 2.0 Specification. http://www.openajax.org/member/wiki/OpenAjax_Hub_2.0_Specification (last access: Oct 2010).

[OpenSocial, 10] OpenSocial. A common Web API for social applications. http://code.google.com/apis/opensocial/ (last access: Oct 2010).

[OSDE, 10] OpenSocial Development Environment (OSDE). An IDE development tool for OpenSocial applications. http://code.google.com/p/opensocial-development-environment (last access: Oct 2010).

[Paterson, 08] Paterson, I., Saint-Andre, P.: XEP-0206: XMPP Over BOSH. Draft Specification. Oct, 2008. http://xmpp.org/extensions/xep-0206.html (last access: Oct 2010).

[People's Daily Online, 07] People's Daily Online: More than 9.5 Mio. Chinese compete in world's largest exam. http://english.peopledaily.com.cn/200706/07/eng20070607_381831.html (last access: Oct 2010).

[Saint-Andre, 04a] Saint-Andre, P.: RFC 3920: Extensible Messaging and Presence Protocol (XMPP): Core. Technical report, Jabber Software Foundation, October 2004.

[Saint-Andre, 04b] Saint-Andre, P.: RFC 3921: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. Technical report, Jabber Software Foundation, October 2004.

[Saint-Andre, 09] Saint-Andre, P., Smith, K., and Tronçon, R.: XMPP: The Definitive Guide - Building Real-Time Applications with Jabber Technologies. O'Reilly Media. April 2009.

[Saint-Andre, 10] Saint-Andre, P.: XEP-0045: Multi-User Chat. Draft Specification. January 2010. http://xmpp.org/extensions/xep-0045.html (last access: Oct 2010).

[Sattari, 06] Sattari, S., Backhaus, W., Henning, K., Sjoer, E.: The Web-based Knowledge Map - A Knowledge Management Tool for Preservation of Vital Expert Knowledge into Higher Education. In: Mit Wissensmanagement besser im Wettbewerb! Tagungsband zur KnowTech 2006, 8. Konferenz zum Einsatz von Wissensmanagement in Wirtschaft und Verwaltung. Hrsg. v. Gronau, N.; Pawlowsky, P.; Schütt, P.; Weber, M.: München, 2006: 229-238.

[Shen, 08] Shen, L., Shen, "The Pervasive Learning Platform of a Shanghai Online College - A Large-Scale Test-Bed for Hybrid Learning Hybrid Learning and Education", In R. Fong, J.; Kwan, R. & Wang, F. L. (ed.), First International Conference, ICHL 2008, Hong Kong, China, August 13-15, 2008, Proceedings, Springer, 2008, 5169, 178-189

[Shindig, 10] Apache Shindig. OpenSocial container reference implementation. http://shindig.apache.org/ (last access: Oct 2010).

[SocialSite, 10] Project SocialSite. https://socialsite.dev.java.net/ (last access: Oct 2010).

[Stevens, 74] W. Stevens, G. Myers, L. Constantine, "Structured Design", IBM Systems Journal, 13 (2), 115-139, 1974.

[Strophe, 10] Strophe. A family of libraries for writing XMPP clients. http://code.stanziq.com/strophe/ (last access: Oct 2010).

[Wolpers, 07] M. Wolpers, J. Najjar, K. Verbert, and E. Duval, „Tracking Actual Usage: the Attention Metadata Approach," in Educational Technology & Society, vol. 10, no. 3, pp. 106-121, 2007.

[Wolpers, 09] Wolpers, M., Memmel, M., Giretti, A.: Metadata in Architecture Education - First Evaluation Results of the MACE System, In Proceedings of the 4th European Conference on Technology Enhanced Learning: learning in the Synergy of Multiple Disciplines, 2009

[XMPP4JS, 10] Xmpp4Js. An XMPP client library written in pure Javascript. http://xmpp4js.sourceforge.net/ (last access: Oct 2010).

[Zhang, 07] Zhang, J. A cultural look at information and communication technologies in Eastern education, In Educational Technology Research and Development, 2007, 55, 301-314