Social Software for Lifelong Competence Development: Challenges and Infrastructure

I. Marenzi, E. Demidova, W. Nejdl, D. Olmedilla and S. Zerr
L3S Research Center, Hanover, Germany

***Abstract*—Within the TENCompetence project we aim to develop and integrate models and tools into an open source infrastructure for the creation, storage and exchange of learning objects, suitable knowledge resources as well as learning experiences. This contribution analyzes the potential of social software tools for providing part of the required functionality, as well as some challenges involved.**

***Index Terms*—Web 2.0; Lifelong Learning; Social Software.**

I. INTRODUCTION

During the last three years, the Web has entered into a second phase, known as Web 2.0. New services and software have transformed the Web from being a predominantly read-only medium to one where anyone can publish and share web contents. Web 2.0 tools promote different types of communication: one-to-one, one-to-many, or many-to-many, synchronous and asynchronous, and can be used to search, share and create different media: from text (Blogs and Wikis) to images in Flickr, audio, podcasting and video in YouTube. Given the information overload that is created by the exponential growth of content on the Web, other tools help learners filter and manage information (social bookmarking and RSS feeds). The use of these services has provided new means to share knowledge, exchange ideas and easily publish work.

Social software at its core is based on supporting individuals to interact socially and to achieve their personal goals, together with people who have similar interests. It works bottom-up: people sign up to a system and form communities through personal choice and actions. Their desire to organize themselves into groups and to collaborate by advancing personal interests contrasts with more traditional approaches where people are placed into organizationally or functionally-defined groups.

In contrast, traditional LMS still approach group membership in a top-down fashion. In current learning environments and in corporate settings, it is hard to imagine a single person acting without some specifically assigned membership (in a class, a working group, a team or a division). Social software will change the traditional way in which learning systems, groupware and other project-oriented collaboration tools work. People start using social software individually; they advance their own biases and connections, and reflect them in social relationships in everyday life. This process is not organized in terms of a single, clearly defined project; rather, it is a people-driven one, in which social interactions are inductive, passing from individual to a group, to other people and other groups. This approach may appear untidy and approximate, but often is a better method towards forming strongly motivated groups and working teams.

In our project context we want to address the following questions:

What happens, if social software is used in formal learning or work environments, and how can it extend the functionalities of traditional learning or work environments? How can the essential elements of social software be incorporated into more conventional software solutions, ultimately transforming learning communication and working collaboration, and which challenges do we have to address to achieve this integration?

The use of Blogs, Wikis, media-sharing services, and other social software, has been shown to create exciting new learning opportunities for people, and to support creation of social networks and communities of practice among company employees [1, 6]. The learner is seen as a participant who is actively engaged through a rich set of interactions within these communities. At the same time, the worker must fulfill three workplace roles: working, learning and collaborating with other colleagues. In this paper, we sketch, in a scenario-oriented way, how people can interact in their working environment to create, search and share knowledge resources [3, 5, 6].

II. TENCOMPETENCE BACKGROUND

TENCompetence addresses the need for flexible and effective lifelong competence development and aims at supporting individuals, groups and organizations by establishing the most appropriate technical and organizational infrastructure, using open source, standards-based, sustainable and innovative technology.

To integrate models and tools for creation, storage and exchange of knowledge resources, in the first project stage the KRSM infrastructure was implemented [2], making information accessible to better support lifelong learning and enhance learning experience. This infrastructure brings together information stored on institutional servers, centralized repositories, locally on learner desktops (by means of P2P technology) and online community-sharing systems like Flickr and YouTube. The KRSM architecture is depicted in Figure 1.

In the next project phase we're extending the integration of Web 2.0 applications to support a variety of scenarios, one of which is described in the next section. Whereas in the current KRSM architecture Web 2.0 applications like Flickr and YouTube are only considered to be information sources for existing services, in the next

project stage we aim to combine functionalities of existing Web 2.0 applications (tagging, bookmarking and commenting) into an integrated LearnWeb 2.0 platform for sharing, discussing as well as for (possibly collaborative) creation of knowledge resources.

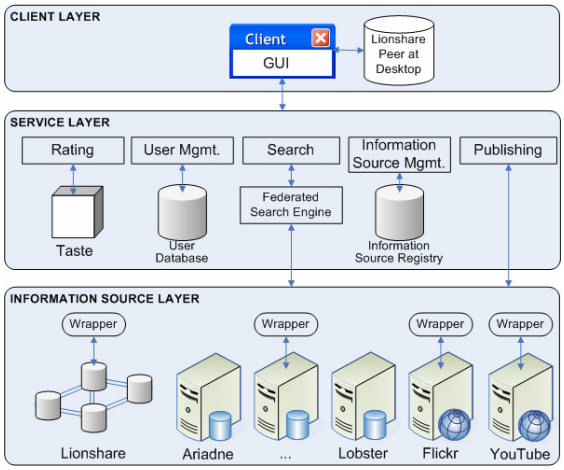

Figure 1. KRSM Architecture

III. SCENARIO AND CHALLENGES

We start with the observation that software development needs to be interpreted and described not only from a technological standpoint, but in terms of potential use. Thus we start with a short scenario and then analyze the challenges and design choices arising from that scenario.

A. Scenario Higher Education / Need for Interoperability

Our main actors in this scenario are ICT technicians who support different projects and people in a university environment. ICT is used at different levels of the organization: work (store and share knowledge resources), teaching (present and provide learning materials), learning (for workers and students). Dynamic changes in the technical infrastructure like hard- and software development require technicians to develop new competences to cope with the continuously evolving environment. Thereby information aggregation and sharing plays a central role to enable speeding up the required competence development.

Each technician works on several tasks, but communication among colleagues is only performed in person or via e-mail, without any synchronization or support. Too often, a technician does not know what the others are doing, even if their work is relevant for her tasks. Resources are stored in different databases, which are neither linked nor interoperable; discussions take place ad hoc and are not stored; best-practice transfer is manual and ad hoc, and is not supported by any system.

Figure 2 depicts some of the technicians' tasks and contextualizes some possible applications of Web 2.0 tools to foster information exchange and thus speed up competence development. For instance, Blogs and Wikis can provide cross-project communication, creating a useful knowledge repository and allowing easier review, reporting and sharing of activities. A technician could browse and subscribe to them through RSS to keep updated on relevant news. Web 2.0 tools can be used to foster interactivity, communication and collaboration.

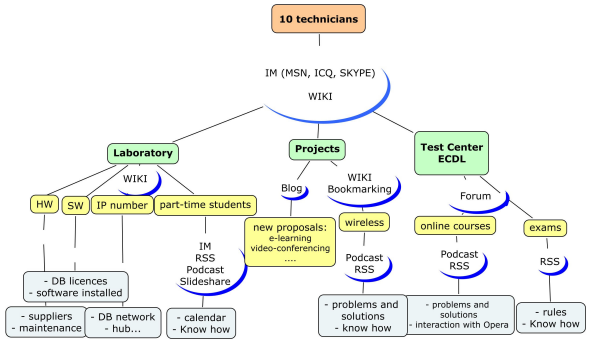

Figure 2. Technicians' Activities

Other examples include the adoption of Instant Messaging to communicate and share information within the technicians' group at lower costs (compared with telephone calls) and the advantage of saving data in a chat. RSS feeds and social bookmarking can help to track changes and news, replacing the staff newsletter. Blogs and social annotation tools support project discussions and development of technical plans. The instant, secure and constant accessibility of data in searchable format, which Blogs provide, can be a huge productivity improvement in sharing information. Wikis and Mind mapping can help in creating a knowledge base of good practices and preparing the agenda for delivering the minutes. Forums can be used to keep track of courses and exams procedures as well as Podcasts as an alternative or supplement to traditional face-to-face training activities, to facilitate IT support for university employees and part-time students.

Also using new means of sharing between different institutions can provide useful insights by adapting new technologies:

- social bookmaking in existing online library services (including online catalogues and online information resources such as e-journals)
- video-conferencing and content sharing to customize university courses
- reuse and sharing knowledge resources and tools between different universities to improve the quality of learning
- Forums to allow students to share their experiences with evolving learning environment.

Traditional information systems like LMS should blend with Web 2.0 applications in order to create new environments that reshape information processes and flows and connect competences. The objective is to allow users to invest as much of the available effort as possible in the production of rich interaction, resulting in an optimal collaborative load. The use of social software applications fosters the sense of community and group motivation, supporting lifelong competence development.

Providing integration and sharing among these different kinds of tools is crucial, though. Already with current ERP solutions, integration of diverse systems turned out to be a challenge. Applications "do not converse"; they do not share data and do not concur to re-use services or applications in a uniform / interoperable way.

With Web 2.0, we have to integrate new application types into this already complicated environment, to provide functionalities for knowledge resource sharing and exchange. Retrieval of heterogeneous knowledge resources among different tools and social network services is still too difficult. We would like to collect relevant information about different knowledge resources, gathering them in an integrated environment from where they can easily be accessed. This should be provided via a distributed and modularized infrastructure, but allow some means of centralized user authentication or Single Sign On (SSO) functionalities, to avoid logging in several times for each tool integrated in our environment.

B. Challenges

Web 2.0 is a challenging environment, in which knowledge resources are distributed among a set of heterogeneous online storage tools, each providing specific functionalities. Whereas each online application supports a limited set of pre-defined tasks (like storage, editing or discussing of resources), our LearnWeb 2.0 integrated environment aims at offering a rich set of functionalities over the whole virtual working space containing the entire set of distributed resources, without unnecessary boundaries.

Existing Web 2.0 tools differ in programming languages, granularity degree of their APIs, and licensing system. Among the great number of available tools, only a few are delivered with an open source license, which allows them to be customized and seamlessly integrated in a centralized environment. Whereas many tools are copyrighted, some of them deliver their API, which allows for integration of their services in LearnWeb 2.0. When neither the source code nor an API is available, they have to be linked as external tools. Our work on LearnWeb 2.0 will address two main challenges in the coming months: a) Integration and Interoperability and b) Identity Management.

C. Integration and Interoperability

Our LearnWeb 2.0 infrastructure aims to provide rich functionalities within a single environment as a combination of services provided by Web 2.0 systems.

Content provided on Wikis, Blogs, Forums, Podcasts and other tools need to be integrated in a way that makes access to these distributed resources as easy as to learning materials in a conventional LMS, and also provide the entire set of required functionalities. Technical integration of different Web 2.0 applications can be performed at different levels. We consider three possible integration degrees: basic, partial and complete. An example of basic integration is linkage of resources provided by one application (for example, a photo in Flickr) from another application (such as a document in Google docs). This basic integration level does not require lots of implementation effort (in fact some existing Web 2.0 applications provide such basic integration by means of links to external resources), but does not really help the user to reduce manual efforts, as all references need to be created manually. A more tight (partial) integration can be achieved by putting one common application on top of the APIs provided by the different Web 2.0 tools. Unfortunately, most of the available APIs are application specific and functionally limited, making even partial integration difficult.

Full integration (the most difficult to achieve) would result in a common system that provides the entire set of functionalities of all applications in an integrated manner. For example, in such a fully integrated system we could drag and drop a Flickr picture to a document written in Google docs. Although at the first glance full integration seems to be the most preferable choice, it needs to be performed in a modular way, preserving the ability of future updates of integrated tools.

One possible approach to achieve such technical integration is to define a set of common interfaces for the core services, such as SQI [7] for search. However, heterogeneous APIs require creating specific wrappers, like SQI wrappers implemented in the KRSM system for YouTube and Flickr integration. A disadvantage of this approach is that such wrappers need to be created for every application to be included. Another problem is that not all required functionality of the Web 2.0 applications can be accessed through their APIs. For instance, the YouTube API does not allow for video upload.

To address this problem in LearnWeb 2.0, we currently investigate a set of core services to be fully integrated and look for suitable tools providing them. In the next phase we will extend the list of supported functionalities by adding new tools using tool- and service dependent degrees of integration. In order to support the core discussion functionality described in our scenario, we will first install freely available Wiki and Forum software on a LearnWeb 2.0 server and then connect services of the other tools like Flickr using their APIs.

Apart from integrating components on an operational level, semantic interoperability has to be provided. Currently, most of the search facilities of the available Web 2.0 applications rely on keyword search using tags. We expect more semantic search features to be added in the future, raising the question of semantic interoperability. Also, some of the tools provide more expressive query languages than the others. Some allow only retrieving single resources, whereas others like Flickr or GroupMe! (http://groupme.org) support resource aggregation.

D. Identity Management

LearnWeb 2.0 will need to provide means of seamless user authentication for every application it integrates. Having to log into a multitude of separate applications would, besides the generated nuisance, slow down search and learning processes significantly.

One of the popular approaches for Single Sign On in a university environment is Shibboleth (http://shibboleth.internet2.edu/) that supports cross-institutional sharing of access controlled web resources. Unfortunately, targeted Web 2.0 tools as well as the users of the LearnWeb 2.0 are typically not a part of any specific organization, reducing the applicability of Shibboleth as a solution candidate.

Another interesting approach for SSO in a Web environment is provided by OpenID (http://openid.net/), an open, decentralized, free SSO system for user-centric digital identity. Using OpenID-enabled sites, Web users do not need to remember traditional authentication tokens (username, password) for every site they want to visit. Instead, they only need to be previously registered on a Website with an OpenID identity provider. As OpenID is

decentralized, any Website – regardless its institutional affiliation – can employ OpenID for users to sign in. OpenID takes advantage of already existing Internet technology (URI, HTTP, SSL, etc.) and employs identities that people have already created for themselves in their blog, photostream, profile page, etc. OpenID does not solve all problems, though. Although many sites already support it, in order to be useful for a fully integrated environment, OpenID needs to be integrated in every included site. Thus selecting OpenID still restricts possible choices for the tools to be integrated.

An important consideration for designing LearnWeb 2.0 is that most users already have their personal accounts with many applications to be integrated. These user accounts can be accessed through the Web interface of the specific tool. Users need to access their own resources, contacts, and bookmarks already available in the Web 2.0 applications through the new integrated environment. One possibility to provide SSO for LearnWeb 2.0 is to keep authentication data required by each application encrypted in a single place (locally by the user or on a trusted server). In this way, the data for a specific application can be decrypted and used to authenticate the user, only requiring the user to provide one password for decryption. The advantages of this approach are its simplicity and independence of the target application.

IV. ARCHITECTURE AND IMPLEMENTATION

To address the challenges described above we are currently implementing a prototype of LearnWeb 2.0 within the scope of TENCompetence project. LearnWeb 2.0 will allow for loose integration of Web 2.0 tools while providing a homogeneous view over the whole distributed learning space.

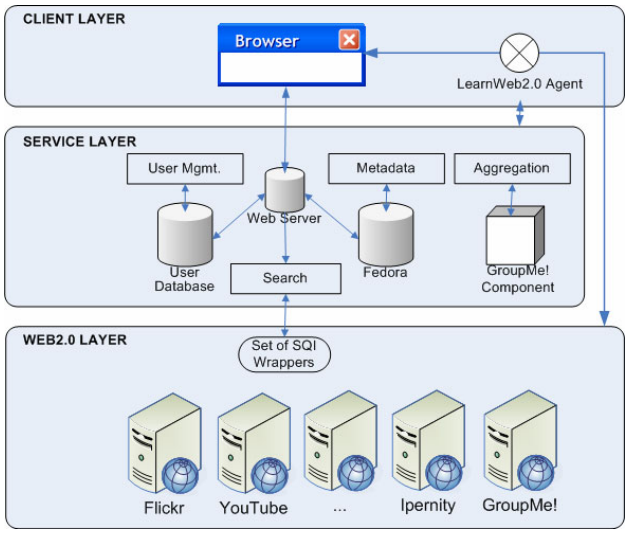

Figure 3. LearnWeb 2.0 Architecture

On their desktop people often place together documents that share similar types, topics or proximity in creation time [4], which can then be used by the file management and search software to assist users in finding and aggregating resources related to a common learning activity. However, on the web users are often forced to distribute related resources across different Web 2.0 applications according to the type of the resources, e.g. put pictures in Flickr, videos in YouTube, and bookmarks in del.icio.us, even if all these resources belong to one and the same learning context. LearnWeb 2.0 will allow users to work in the native environment of Web 2.0 tools while monitoring such time correlated activities using an agent integrated in the user's web browser. This agent will support user-defined as well as automatic activity-based aggregation of resource references keeping track of the user's learning activities in the distributed environment. The user can then access LearnWeb 2.0 application to work directly with the collected metadata, explicitly add external resources from a set of more closely integrated tools as well as share single resources and resource groups with other users.

We envision LearnWeb 2.0 as a 3-layered extensible open source infrastructure consisting of client, service and Web 2.0 layers as presented in Figure 3.

A. *Web 2.0 Layer*

The Web 2.0 layer consists of a set of loosely integrated Web 2.0 applications. Selected Web 2.0 applications can be more closely integrated in our infrastructure by means of implementing an SQI wrapper [7] providing a common search interface. The wrapper translates LearnWeb 2.0 queries and search results to comply with the API of the target tool. Currently we provide such wrappers for YouTube and Flickr.

B. *Service Layer*

The service layer of LearnWeb 2.0 contains components for user and resource metadata management, resource aggregation and search. Components in this layer will interact with the LearnWeb 2.0 agent in the browser as well as with selected Web 2.0 applications.

The User Management component contains user profiles as well as user login data to Web 2.0 applications in an encrypted form to allow for SSO.

The Metadata Management component incorporates metadata of the resources provided by the Web 2.0 layer. Resource metadata like title and description, as well as comments, tags and ratings from LearnWeb 2.0 users are stored by this component for each resource. Currently, metadata storage is implemented using Fedora repository.

The Aggregation component, which is a part of our future work, will provide facilities for grouping resources related through a common learning activity. To allow for resource aggregation on the server side, LearnWeb 2.0 will make use of the GroupMe! platform. GroupMe! is a Web 2.0 application which supports aggregation of external resources into user-defined groups as well as tagging and sharing of the resulting groups to help users in categorizing resources. It provides visualization of the created groups as well as search functionalities.

Finally, the Web server component provides access to the LearnWeb 2.0 services. It supports the following functions:

- Managing the user profile including access data to external Web 2.0 applications for SSO
- Managing knowledge resources, including search, displaying and editing resource metadata
- Bookmarking, tagging, commenting and rating knowledge resources
- Adding external resources via a search interface. The search interface allows for keyword search on metadata stored within LearnWeb 2.0 as well as direct search on Web 2.0 sources via an API

C. *Client Layer*

Currently, the LearnWeb 2.0 client layer is represented through a web client, which runs in a web browser and provides quick access to the LearnWeb 2.0 services. In our future work we plan to implement a LearnWeb 2.0 agent as a browser extension. This agent will allow using LearnWeb 2.0 resource aggregation facilities with arbitrary Web 2.0 sources. To support explicit aggregation of resources, the agent will recognize drag and drop events. On the drag event over a resource which can be added to a group, a window of the GroupMe! component will appear in the left bottom corner of the page. This window will provide an overview over user-defined groups enabling user to add a resource to a group by the drop event. This interface is presented in Figure 4. To allow for implicit activity-based resource aggregation, the LearnWeb 2.0 agent will monitor user interaction with Web 2.0 applications and provide this information to the aggregation component of the service layer.

D. *Using LearnWeb 2.0*

As a newly hired ICT technician, Alice is requested to perform an installation of the Stud.IP LMS system (www.studip.de) on the university server. This installation requires experience with the PHP programming language as well as with Linux server administration. Alice is experienced with Linux; her expertise with respect to PHP is very low, however.

Alice uses LearnWeb 2.0 to collect necessary information about PHP and Stud.IP. As a new LearnWeb 2.0 user, Alice creates a LearnWeb 2.0 account. Optionally, she can install the LearnWeb 2.0 plug-in in her favorite browser. Now she can access the user management service of LearnWeb 2.0 to edit her preferences. For instance, she can provide login data for her favorite Web 2.0 applications to allow for Single Sign On later on. Alice can also specify her preferences for running LearnWeb 2.0 agent.

Figure 4. LearnWeb 2.0 User Interface

To start a PHP learning activity using Web 2.0 tools, Alice logs in to the LearnWeb 2.0 page. Her encrypted access data for the selected Web 2.0 applications are downloaded from the LearnWeb 2.0 server and decrypted on the client side. Her login to the specified Web 2.0 applications is processed automatically, such that Alice can directly access a PHP video tutorial on YouTube and start a learning activity. In case LearnWeb 2.0 agent is installed in her browser, Alice can identify an important learning resource to the agent by an explicit drag and drop event while working directly with various Web 2.0 tools. As Alice drags a video resource, the LearnWeb 2.0 agent offers a selection of groups to add the resource reference. Alice creates a new group "LMS" and drags the video into it. Additionally, her activities can be automatically collected by the agent. Finally, Alice can access LearnWeb 2.0 page to obtain an overview over the groups of distributed learning resources as well as add related resources by search. She can browse search results and look up the details of a particular resource. Alice finds a group of related resources created by a colleague from a partner university who already solved a similar task and adds them to her LMS group. She can comment, tag and rate resources. Finally, she shares the LMS group with other ICT technicians of her university.

V. CONCLUSIONS AND FUTURE WORK

Internet has changed the way people acquire and share knowledge. Web 2.0 infrastructures will change the way people exchange knowledge and interact. In this paper we used one scenario to find and discuss some challenges for integrating social software tools in our LearnWeb 2.0 infrastructure, and sketched the architecture we are currently implementing. So far, interoperability aspects considered in our work are resource centric. In future we plan to consider additional aspects of community building and sharing in an integrated Web 2.0 environment to further support shared learning experiences.

REFERENCES

[1] Budznik, J., Hammond, K. Anticipating and Contextualizing Information Needs. In: Proc. of the 62nd Annual Meeting of the American Society for Information Science, 1999.

[2] Demidova, E., Kärger, P., Olmedilla, D., Ternier, S., Duval, E., Dicerto, M., Mendez, C., Stefanov, K. Services for Knowledge Resource Sharing & Management in an Open Source Infrastructure for Lifelong Competence Development. ICALT, 2007.

[3] Eraut, M. Informal learning in the workplace. Studies in Continuing Education 26(2), 247–273, 2004.

[4] Nardi, B. and Barreau, D. "Finding and reminding" revisited: appropriate metaphors for file organization at the desktop. ACM SIGCHI Bulletin 29, 1 (1997), 76–78.

[5] Rosenberg, M.J. E-Learning: Strategies for Delivering Knowledge in the Digital Age. McGraw-Hill, 2001.

[6] Schmidt, A., Braun, S. Context-Aware Workplace Learning Support: Concept, Experiences, and Remaining Challenges. In Proc. of EC-TEL 2006.

[7] Simon, B., Massart, D., Assche, F., Ternier, S., Duval, E., Brantner, S., Olmedilla, D., Miklós, Z. A Simple Query Interface for Interoperable Learning Repositories. In Proc. of the Workshop "Interoperability of Web-Based Educational Systems", 2005.

AUTHORS

Ivana Marenzi received her master's degree in Science Technology and Media at the University of Pavia (Italy) where she currently works as a technician in the ICT department. Her interests include Internet, multimedia and communication technology in support of education.

Elena Demidova graduated from the University of Osnabrück, Germany. Her research interests include database usability and information extraction.

Prof. Dr. Wolfgang Nejdl has been full professor of computer science at the University of Hannover since 1995. He is heading the L3S Research Center (http://www.l3s.de/), and does research in the areas of

semantic web technologies, search and information retrieval, databases and technology enhanced learning.

Dr. Daniel Olmedilla holds a PhD from Universidad Autónoma de Madrid, Spain. His research interests include topics in the areas of Semantic Web, and Security and Trust Management.

Sergej Zerr holds a M.Sc. degree from the University of Osnabrück, Germany. His research interests include confidential information retrieval and database security.

The work on this paper has been partially sponsored by the TENCompetence Integrated Project. Contract 027087.

Manuscript received 09 Mai 2008. This article was modified from a presentation at the 4th Competence Open Workshop in Madrid, Spain, April 2008. Published as ubmitted by the authors.