

Proceedings of the First International Workshop on Mashup Personal Learning Environments

Citation for published version (APA):

Wild, F., Kalz, M., & Palmér, M. (2008). *Proceedings of the First International Workshop on Mashup Personal Learning Environments*.

Document status and date:

Published: 29/09/2008

Document Version:

Peer reviewed version

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:

<https://www.ou.nl/taverne-agreement>

Take down policy

If you believe that this document breaches copyright please contact us at:

pure-support@ou.nl

providing details and we will investigate your claim.

Downloaded from <https://research.ou.nl/> on date: 15 Jul. 2023

Open Universiteit
www.ou.nl





Mash-Up Personal Learning Environments (MUPPLE'08)

Fridolin Wild, Marco Kalz, Matthias Palmer (Eds.)

*Workshop in Conjunction with the
3rd European Conference on Technology-Enhanced
Learning (ECTEL'08): Times of Convergence*



Please refer to these proceedings as

F. Wild, M. Kalz, M. Palmér: Mash-Up Personal Learning Environments. Proc. of 1st Workshop MUPPLE'08, Maastricht, The Netherlands, September 17, 2008, CEUR Workshop Proceedings, ISSN 1613-0073, online CEUR-WS.org/Vol-388/.

Copyright © 2008 for the individual papers by the papers' authors. Copying permitted for private and academic purposes. Re-publication of material from this volume requires permission by the copyright owners.

Fridolin Wild, Marco Kalz, Matthias Palmér (Eds.)

***Mash-Up Personal
Learning Environments
(MUPPLE'08)***

Proceedings of the MUPPLE'08 Workshop in
conjunction with the EC-TEL 2008, Maastricht, The
Netherlands, September 17, 2008.

Organisers

Fridolin Wild (Vienna University of Economics and
Business Administration, Austria)
Marco Kalz (Open University, The Netherlands)
Matthias Palmér (University of Upsala, Sweden)

Programme Committee

Abelardo Pardo (University Carlos III de Madrid, Spain)
Ajith Ranabahu (Wright State University, Ohio)
Dai Griffith (University of Bolton, UK)
Denis Gillet (EPFL, Switzerland)
Effie Law (University of Leicester, United Kingdom)
Fabrizio Giorgini (Giunti Labs, Italy)
Felix Mödritscher (Vienna University of Economics
and Business Administration, Austria)
Graham Atwell (Pontydygu, United Kingdom)
Gytis Cibulskis (Kaunas Technical University, Lithuania)
Mart Laanpere (Tallinn University, Estonia)
Martin Wolpers (Fraunhofer FIT, Germany)
Mohamed Amine Chatti (RWTH Aachen, Germany)
Nikos Karacapilidis (University of Patras, Greece)
Scott Wilson (University of Bolton, United Kingdom)
Stéphane Sire (EPFL, Switzerland)
Tony Hirst (Open University, UK)

Editorial

A change in perspective can be certified in the recent years to technology-enhanced learning research and development: More and more learning applications on the web are putting the learner centre stage, not the organisation. They empower learners with capabilities to customize and even construct their own personal learning environments (PLEs). These PLEs typically consist of distributed web-applications and services that support system-spanning collaborative and individual learning activities in formal as well as informal settings.

Technologically speaking, this shift manifests in a learning web where information is distributed across sites and activities can easily encompass the use of a greater number of pages and services offered through web-based learning applications. Mash-ups, the 'frankensteining' of software artefacts and data, have emerged to be the software development approach for these long-tail and perpetual-beta niche markets. Core technologies facilitating this paradigm shift are Ajax, javascript-based widget-collections, and microformats that help to glue together public web APIs in individual applications.

In a wide range of European IST-funded research projects such as iCamp, LTfLL, LUISA, Palette, Prolix, and TENcompetence a rising passion for these technologies can be identified. This workshop therefore serves as a forum to bring together researchers and developers from these projects and an open public that have an interest in understanding and engineering mash-up personal learning environments (MUPPLEs).

The aim of this workshop is to bring together the various research and development groups in technology-enhanced learning that currently focus on the development of the next generation learning environments – learning environments that put the individual centre stage and empower learners with design capabilities by deploying modern mash-up principles to establish system-spanning interoperability.

As this approach is rather young, the workshop sought to attract both research results and work in progress in order to chart out the current state-of-the-art of MUPPLEs in TEL and to define main enablers and future challenges. Naturally, it serves as a forum for establishing new collaborations.

Summary of the Contributions

The long paper by Mödritscher, Wild, and Sigurdarson on '*Language Design for a Personal Learning Environment Design Language*' sketches a vision on learning environment design being the key to contemporary transcompetence acquisition by

treating learning environments as manipulable outcomes of learning activities that are designed for emergence. They draft a domain-specific design language called 'learner interaction scripting language' (LISL) and investigate how compliant this language is with postulated (end-user development) design principles and guidelines.

Asensio-Pérez, Bote-Lorenzo, Vega-Gorgojo, Dimitriadis, Gómez-Sánchez, and Villasclaras-Fernández propose an architecture in their short paper *'Adding mash-up based tailorability to VLEs for scripted Collaborative Learning'* and report on first prototype where a central integration manager parses IMS-LD scripts to subsequently configure, in interplay with an educator, those tool instances deployed in the foreseen activities to deliver individualised learning experiences – thereby adding openness and tailorability to otherwise static, monolithic virtual learning environments.

Fuente Valentín, Leony, Pardo, and Delgado Kloos report in their long paper *'Mashups in Learning Design: pushing the flexibility envelope'* how IMS Learning Designs can be combined with the mash-up configuration language LISL. The paper describes how the LISL language is utilized to bring the flexibility of the mash-up world into a particular learning environment, namely GRail, i.e. the learning design run-time environment for the .LRN platform. The concepts of micro and macro scripts are brought in from the area of CSCL to find the right balance of flexibility.

Wilson, Sharples, and Griffiths outline in a long paper on *'Distributing education services to personal and institutional systems using Widgets'* an open standards approach for creating and embedding widgets. The approach discussed is based on the W3C widget specification extended with collaboration features. The approach is demonstrated through the implementation of a widget server, called Wookiee, as well as two proof-of-concept container applications in Wordpress and Moodle. The paper then shortly discusses how this kind of approach can allow institutionally maintained services to appear in Personal Learning Environments.

With their short paper, Sire and Vagner sketch a vision for *'Increasing Widgets Interoperability at the Portal Level'*. The discussion is focused on an API for capturing widget-to-widget communication in a web portal. The paper identifies two communication scenarios: human-initiated drag & drop and communication-initiated based on widget state changes.

Casquero, Portillo, Ovelar, Romo, and Benito outline in the short paper *'iGoogle and gadgets as a platform for integrating institutional and external services'* a framework on how generic mash-up platforms like iGoogle can be instrumentalized to pre-build individual PLEs. Furthermore, they discuss which generic and which institutional services could be mapped into these.

Bogdanov, Salzmann, El Helou, and Gillet discuss in their short paper *'Social Software Modeling and Mashup based on Actors, Activities and Assets'* a conceptual model for web applications. With the help of it, successfully modelled and mapped web applications can be mashed-up, i.e., combined both visually and on the data level. Additionally, the paper illustrates the deployment of this model with an example mash-up container, i.e., the eLogbook environment integrating.

Ebner and Palmér propose '*A Mashup-friendly Resource and Metadata Management Framework*'. The long paper focuses on establishing a mash-up-friendly resource and metadata management with an appropriate level of generic functionality, with which the development of targeted tools becomes a matter of user interfaces and specialisation. Already existing standards are looked upon and it is explained why they are not generic enough. An implementation of the proposed framework is presented along with a tangible use case.

Soylu, Kuru, Wild, and Mödritscher propose a '*A Learning Object Harvesting Model and a Sample Application*' in a long paper which introduces a framework for automatically extracting a proposed learning object microformat embedded in web pages with the help of XSL transformations following the W3C GRDDL specification. A simple query interface (SQL) that accepts SPARQL queries is provided in a prototype for the retrieval of the learning objects.

Vuorikari outlines an approach for '*Consolidating collections of learning resources using APML*' in a short paper, backed up by the findings in a case study on teachers' use of social bookmarking tools. This early work sketches a rough vision on how contextualized attention meta-data can facilitate a higher level integration and syndication of scattered bookmark collections.

Weber, Thomas, and Ras analyse in their long paper on '*Investigating the Suitability of Mashups for Informal Learning and Personal Knowledge Management*' how well mash-ups can be utilized for personal, informal knowledge management. Therefore, common mash-up features are mapped against PKM objectives that are grouped into seven information skills.

Attwell, Bimrose, Brown, and Barnes analyse in the long paper '*Maturing learning: Mash up Personal Learning Environments*' why and how a personal learning environment should support knowledge maturing and development. They outline a scenario and elaborate, with the help of it, a functional taxonomy of tasks to be supported. Deliberations on development approaches round up the paper.

Acknowledgements

This workshop is supported by the European Commission as a concertation activity between especially the European Projects iCamp, LUISA, LTfLL, Palette, Prolix, and TENcompetence. We especially would like to thank the members of our programme committee listed below.

*Fridolin Wild,
Marco Kalz,
Matthias Palmér*

Table of Contents

Editorial

Fridolin Wild, Marco Kalz, Matthias Palmér	1
--	---

Language Design for a Personal Learning Environment Design Language

Felix Mödritscher, Fridolin Wild, Steinn Sigurdarson	5
--	---

Adding mash-up based tailorability to VLEs for scripted Collaborative Learning

Juan I. Asensio-Perez, Miguel L. Bote-Lorenzo, Guillermo Vega-Gorgojo, Yannis A. Dimitriadis, Eduardo Gomez-Sanchez, Eloy D. Villasclaras-Fernandez	14
---	----

Mashups in Learning Design: pushing the flexibility envelope

Luis de la Fuente Valentin, Derick Leony, Abelardo Pardo, Carlos Delgado Kloos	18
---	----

Distributing education services to personal and institutional systems using Widgets

Scott Wilson, Paul Sharples, Dai Griffiths	25
--	----

Increasing Widgets Interoperability at the Portal Level

Stephane Sire, Alain Vagner	33
-----------------------------------	----

iGoogle and gadgets as a platform for integrating institutional and external services

Oskar Casquero, Javier Portillo, Ramón Ovelar, Jesús Romo, Manuel Benito	37
--	----

Social Software Modeling and Mashup based on Actors, Activities and Assets

Evgeny Bogdanov, Christophe Salzmann, Sandy El Helou, Denis Gillet	42
--	----

A Mashup-friendly Resource and Metadata Management

Hannes Ebner, Matthias Palmér	48
-------------------------------------	----

A Learning Object Harvesting Model and a Sample Application

Ahmet Soylu, Selahattin Kuru, Fridolin Wild, Felix Mödritscher	57
--	----

Consolidating collections of learning resources using APML

Riina Vuorikari	65
-----------------------	----

Investigating the Suitability of Mashups for Informal Learning and Personal Knowledge Management

Sebastian Weber, Ludger Thomas, Eric Ras	69
--	----

Maturing learning: Mash up Personal Learning Environments

Graham Attwell, Jenny Bimrose, Alan Brown, Sally-Anne Barnes	78
--	----

Language Design for a Personal Learning Environment Design Language

Felix Mödritscher, Fridolin Wild, Steinn Sigurdarson

Institute for Information Systems and New Media,
Vienna University of Economics and Business Administration,
Augasse 2-6, 1090 Vienna, Austria
{felix.moedritscher,fridolin.wild,steinn.sigurdarson}@wu-wien.ac.at

Abstract: Approaching technology-enhanced learning from the perspective of a learner, we foster the idea of learning environment design, learner interactions, and tool interoperability. In this paper, we shortly summarize the motivation for our personal learning environment approach and describe the development of a domain-specific language for this purpose as well as its realization in practice. Consequently, we examine our learning environment design language according to its lexis and syntax, the semantics behind it, and pragmatical aspects within a first prototypic implementation. Finally, we discuss strengths, problematic aspects, and open issues of our approach.

Keywords: Personal Learning Environments, Language Design, Learner Interactions, End-User Development

1 Introduction

New economic-technological trends, subsumed under the term Web 2.0, have started to influence the understanding for and approach to technology-enhanced learning. Additionally current research and development is focus on extending the classical didactic learning management systems by emphasizing functionality supporting the learner, e.g. by dealing with the design of personal learning environments, coping with social networking functions and recommendation services, or examining collaboration and community-boosting features in learning networks. In sum, networked communities are expected to be beneficial for lifelong learning [1].

Also heading into this direction, we summarized shortcomings of instructional design theories and adaptive educational technologies in [2] and came up with our approach; **Mash-UP Personal Learning Environments (MUPPLE)**. Hereby, we build upon three important concepts of learning environment design:

- First, we prefer the idea of ‘*learning to learn*’ in connection with learning content, to ‘transferring’ domain-specific knowledge. Thus we apply a simple and domain-independent model of learning activities which consists of a set of (learner) actions bound to objects (artifacts or outcomes) and tools. Furthermore, we emphasize the *acquisition of transcompetences* (i.e. social, self, and methodological competence) in addition to content competence.

- Second and consequently, we consider the *learning environment* an important part of the *learning outcome* as opposed to an instructional condition. Therefore, a learner *designs her learning environment* by establishing a network of people, artifacts, and tools (manually or with the support of personalization services) and interacting with that environment.
- Third and finally, we consider emergence of behavior as an unavoidable and natural phenomena of complex socio-technical systems. By emergent behavior we mean that the observable dynamics show unanticipated activity, surprising in so far as the participating systems have not been designed for it specifically (they may even not have intended it). *Designing for emergence* is, in our view, more powerful than rule-based personalization, as the models involved are simpler while achieving the same effect.

As a consequence of the application of these three concepts, we decided to follow a learner-centered approach to personal learning environments. In accordance with the paradigm of end-user development depicted in [3], we have been developing a domain-specific language for learning environment design which we named **learner interaction scripting language (LISL)**. In the following section we briefly summarize the principles of language design. In section 3 we describe LISL, a first prototypic implementation, and its utility, before our approach is discussed.

2 Principles for Language Design

In order to achieve end-user developed emergence of learning behavior (MUPPLE), the scripting language (LISL) needs to take into account not only traditional software engineering issues, but also those of end-user development. We consider LISL to be the underlying model for the success of the MUPPLE approach.

Hoare meant a programming language to support three tasks within the software development process [4]: (1) program design, (2) programming documentation, and (3) program debugging reasons. Therefore, he made up five language design principles to achieve these task-specific requirements: First, *simplicity* of a language is critical for reducing the complexity of programming tasks. Second, *security* features should not be removed when going into production, so that programming errors are not hidden away from the end-user for reliability reasons. Third, *fast translation* of program code is of particular interest for prototyping or debugging, which already reminds of younger trends like scripting of command-line interpreters. Fourth, *efficient object code* is required, particularly if following the design for emergence paradigm for end-users developing and optimizing their learning environments. Fifth, compiling and processing programs should increase *readability*, e.g. by automated intention, understandable commands, filling in defaults, etc.

Additionally to these historical principles, younger trends focus on derivations of programming languages. For instance, Paige and his colleagues address modeling language design and divide them into four task-related classes [5]: (1) architectural description, (2) behavioral description, (3) system documentation, and (4) forward and backward generation. For these kinds of modeling languages the following design principles are recommended: First, a language should be *simple* to have it small and

memorable. Second, it should satisfy the principle of *uniqueness* (orthogonality) which means that every concept of interest can be expressed in exactly one way and, thus, the language is defined by a small number of powerful features. Third, the principle of *consistency* refers to a purpose to the design of the language, so that the features included or to be added to the language must further this purpose. Fourth, *seamlessness* contributes to producing maintainable software, i.e. through generating code from models by allowing the mapping of abstractions in the problem space to implementations in the solution space without changing the notation.

Fifth, the principle of *reversibility* requires that changes made during one stage of the development lifecycle can be automatically reflected back to earlier stages. Sixth, *scalability* ensures that a language should be useful to model systems with a few components and inter-relations, and systems with thousands of components and inter-relations, which is achievable by providing a concise mechanism for describing the fundamental abstractions for their problem domain. Seventh, *supportability* demands that a language can be easily produced by hand, so that it can be used, by humans, for writing or drawing models. Eighth, a modeling language must support the production of *reliable* programs which meets its specification and reacts appropriately whenever unexpected or erroneous input is given. Ninth, the principle of *space economy* states that models should take up as little space on a printed page as possible but without sacrificing the understandability of the language, which empowers the production of concise models and decreases the maintenance efforts.

Concerning language design methodologies, Van Deursen et al report about a design approach to domain-specific languages. Basically, a domain-specific language is defined as “*a programming language or executable specification that builds on appropriate notations and abstractions, offering expressive power focused on, and usually restricted to, a particular domain*” [6]. Thus, such a language is expected to be based in an application domain. They suggest building an algebraic model behind the language, consisting of normalization, variability, and satisfaction rules. This algebraic model should be derived from an analysis of the domain by applying feature descriptions, i.e. with feature diagrams or a textual notation.

All in all, this short inspection of literature already indicates that the design of a new language is a complex task. Considering the basic principles noted by Hoare nearly four decades ago is, however, a solid first step to designing a usable programming language. Moreover, design guidelines for modeling languages are of interest for our learning environment design approach. In addition, we consider our scripting approach to be highly related to a domain-specific language and, thus, a methodological approach like the one by Van Deursen et al is relevant. Finally, human-computer interaction research started to influence software engineering, which lead to the concept of end-user development and evolves software engineering from making systems that are “*easy to use*” to making systems that are “*easy to develop*” [3]. Again, this new way of user-driven development of the working environment which is used for interactions in shared spaces and networked communities requires special design guidelines, which amongst others, are elaborated by [7]. We consider end-user development as well as these guidelines to be of importance to our MUPPLE approach and explain them in the upcoming section when describing the development process of our learner interaction scripting language (LISL).

3 LISL and its Interpreter

With respect to the principles outlined in the last section, we decided to build up a human-like language for learning environment design and, additionally, focus on user-related issues like learnability, efficiency, simplicity, readability, uniqueness, seamlessness, reversibility, and supportability. More technical issues, e.g. security, fast translation or efficient object code, are considered to be secondary. In this section, our learner interaction scripting language (LISL) is described in detail.

The **lexical and syntactical structure** is kept very simple. The tokenizer splits each line of code into tokens separated by white spaces. Single or double-quoted constructs are considered to be one token. Moreover, tokens are case-sensitive, and identifiers are restricted to alphanumeric strings (beginning with a letter from the alphabet) and might be quoted. Syntactically we realized only a few constructs with this domain-specific language.

```

1> define action Compose with url http://[...]?action=create
2> define action Browse
3> define action Bookmark
4> define object 'self-description'
5> define object 'self-descriptions of peers' with url http://[...]/peers
6> define object 'selected self-descriptions'
7> define tool VideoWiki with url http://videowiki.icamp.eu
8> define tool Scuttle with url http://scuttle.icamp.eu
9> connect tool VideoWiki with tool Scuttle
10> Compose 'self-description' using VideoWiki
11> Browse 'self-descriptions of peers' using VideoWiki
12> Bookmark 'selected self-descriptions' using VideoWiki
13> drag tool VideoWiki

```

Fig. 1. Example LISL script consisting of three actions, three objects, two tools, as well as one connect, three action and one learner interaction statements

As indicated with the example activity in Fig. 1, we differentiate between the following statement types:

- ‘Define’ statements (cf. lines 1 to 8) are used to initialize the mash-up personal learning environment, i.e. to declare actions, objects, and tools available in one mashup page. The actor is considered to be the current user. Each defined entity stands for a variable in the ‘program’, whereby an action and an object can have an optional URL, while the tool always has one. These kinds of constructs always start with the command ‘define’ followed by the type of entity (‘action’, ‘object’, ‘tool’) and a unique identifier. If a URL can be specified, the keyword ‘url’ followed by the URL itself is expected. Additional tokens are ignored, so that constructs like ‘with the url http://[...]’ are valid.
- ‘Connect’ statements (line 9) are necessary for interoperability reasons, so that tool combinations can be utilized for one action. Syntactically, such a statement allows combining two tools with each other, so that data is sent from one to the other. What this exactly means and how we intend to realize this kind of interoperability, will be clarified when explaining the semantics of this statement.

- ‘Action’ statements (lines 10 to 12) comprise the actions to be performed by a learner and consist of three parts: (a) the action, (b) the object, and (c) the tool. Generally, the first token, plain or quoted, stands for the (user-definable) action, while the second one represents the object. The tool has to be specified right after the keyword ‘using’, other tokens are ignored. If one of the elements of such a statement can not be resolved, an error with an explanation is displayed.
- ‘Learner interaction’ statements (line 13) materialize the user interactions with the learning environment, i.e. they describe if the learner navigates between two different learning tools or she rearranges the application on her screen.

Referring to the end-user development guidelines (DG) highlighted in [7], we consider the three syntactical ones as fulfilled for LISL: (DG 3.1) Syntactical errors are hard to be made due to adequate error messages and visualization techniques by the LISL interpreter (shown later). (DG 3.2) It is even impossible to make syntactical errors, as incorrect statements are not executed and, thus do not halt the execution of the activity. (DG 3.3) We tried to use objects as language elements in the way that one action statement comprises a tool within the learning activity. Such a tool can be drag and dropped as an ‘object’ (window) within a web application mashup [8].

Going on with the **semantics behind LISL**, we build up a simple, comprehensible model of learning activities, as shown in Fig. 2. Precisely, a learning activity consists of a set of (learner) actions which are bound to one object and involve at least one tool. These actions specify typical learner interactions within the activity, whereby they can be defined by learners. The ones with a specific URL are useful to perform a standard operation within a learning tool, e.g. creating a new Wiki page. An action without an URL delegates the initialization of the web application to the object or, if it does not have an URL, to the URL of the first tool after the ‘using’ keyword.

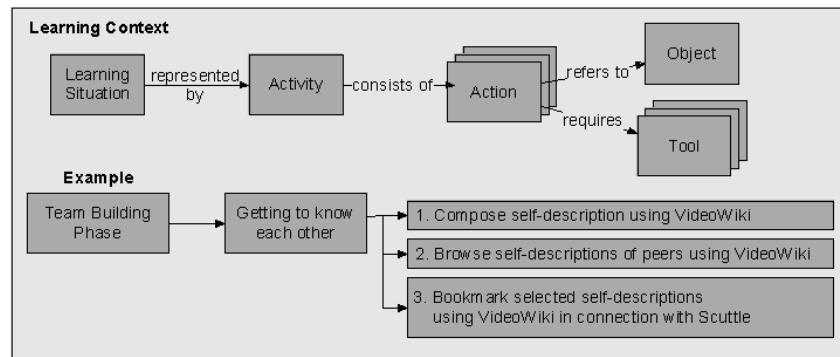


Fig. 2. LISL’s semantic model of learning activities consisting of actions-object-tool triples

In analogy to variable declarations, an object can, but must not be defined with a specific URL. An initialized object means that the outcome has a pre-defined pointer to a specific artifact within a tool. An object without a URL, however, is indicating that the ‘value’ is assigned dynamically as a learner goes through the actions, e.g. by creating a digital artifact using an action with a URL. In our example described in the last two figures, for instance, the very first action creates a VideoWiki recording

which is assigned to the ‘self-description’ object after finalizing the action. Binding a tool combination to one action requires a certain degree of interoperability of the learning tools. As a precondition for our MUPPLE approach, we build upon distributed feed networks and an API we have been developing within the iCamp research project [9]. If this kind of interoperability is not supported, an error is given.

We view the semantic guidelines for end-user development [7], as an affirmation of our scripting language approach. First, LISL is a domain-specific language for end-user development, comprised of human-like, understandable constructs like the action statements (DG 4.1). Second, LISL also includes meta-domain orientation for general end-user development, i.e. by having neutral, domain-independent statements, like ‘define’ or ‘drag’, wherever abstraction from the TEL domain is possible (DG 4.2). Finally, the LISL runtime, MUPPLE supports semantic annotations (DG 4.3) in two different ways. On the one hand, a graphical user interface encapsulates the scripting activity itself, so that learners can use web-based widgets to ‘program’ their personal learning environment. On the other hand, semantic annotations can be realized by recommending action-object-tool bindings of peers.

From the point of view of **pragmatics**, we implemented an interpreter as well as a web-based interface for LISL. Technologically, this prototype is realized as part of the OpenACS framework, an open source toolkit for building scalable, community-oriented web applications (cf. <http://openacs.org>). The LISL interpreter is written in the programming language Tcl, precisely the object-oriented extension named XoTcl (<http://media.wu-wien.ac.at>), and is part of the Mupple package which itself is based on the XoWiki module (<http://openacs.org/xowiki>). The Mupple package, including the source files, can be retrieved from the iCamp code repository at Sourceforge (<http://sourceforge.net/projects/icamp>).

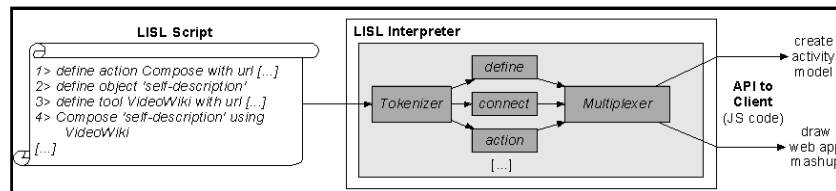


Fig. 3. Functional diagram of the LISL interpreter including a tokenizer, interpreter units for each statement type, and a multiplexer controlling different APIs

Fig. 3 visualizes how the LISL interpreter works beneath the surface. Going through the script line by line, the input is tokenized and, according to the first token, processed by the adequate interpreter unit. Each of these units is representing one specific statement type and produces output for different purposes. Our current implementation differentiates between two output channels, one for creating the learning activity model on the client-side, the other one for creating and updating the mashup of the web-based learning tool. Hereby, the interpreter calculates the latest version of the model and the mashup space, before this information is transferred via a JS-based API to the browser. Updates are made incrementally, so that learners (facilitators and peers!) can execute single lines of code at any time.

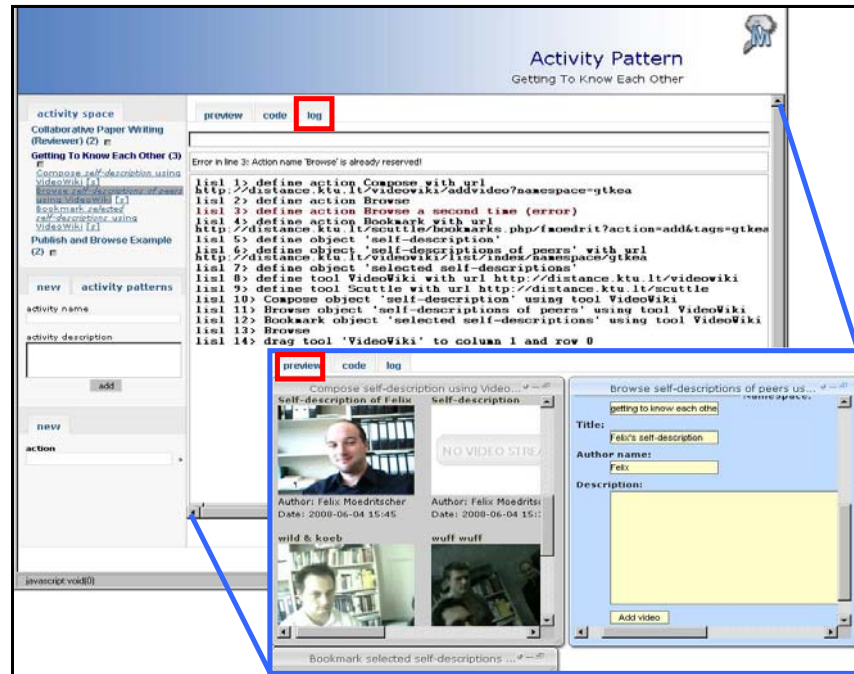


Fig. 4. MUPPLE page for the activity 'Getting To Know Each Other'; the tab 'log' shows the LISP script of our example executed by the interpreter (red indicates an error), while the tab 'preview' (blue border) visualizes the web application mashup and provides web-based widgets

As shown in Fig. 4, our MUPPLE prototype displays a learning activity in the form of a page which is the central user interface of the personal learning environment. At the top, the type (activity or activity pattern) and the title of the page are presented. On the left, all activities of a user are listed. Clicking on an activity folds out the list of its actions and loads it into the content area. Below there are various functions for creating new activities (blank or from a pre-defined pattern), deriving new patterns from the current activity, or starting a new action. For starting a new action MUPPLE supports the learner by recommending action-object-tool triples.

Within the tab 'log', the content area displays the materialization of the learner interactions; the LISP code of the current activity page. By switching the view to the tab 'preview' (the area with blue boarder), learners see the web application mashup space generated from the executing LISP code. The code itself is created either by manual scripting or by logging the interactions with the web-based widgets and stored within the MUPPLE page. On executing the script, the semantic model is built up, and the web applications are launched in windows on the mashup space. Now, learners can work with the web-based control widgets, whereby their interactions are materialized by appending new LISP statements to the MUPPLE page. On returning to a page, its last state will be restored, which we consider as important for scrutable behavior of MUPPLE. Furthermore, any part of the learning activity is subject to

adaptation by the user (facilitators and peers) for controllability reasons. Technically, LISL is similar to AppleScript (<http://www.apple.com/applescript>), but it does not limit the user to automate interactions with an application. Instead, it enables learners to reflect their learning process and to collaborate in learning networks.

Again, we analyze LISL according to the guidelines for the pragmatics of an end-user development language [7]. First, the scripting approach and the web-based command-line interpreter seem to be ideal for supporting incremental development of a personal learning environment (DG 5.1) by being able to see and modify the LISL script of a page (tabs ‘log’ and ‘code’). While our approach does not directly facilitate decomposable test units (DG 5.2), the interactive design allows learners to instantly see the systemic reactions to new or modified code, thus simplifying testing. As the underlying storage layer also supports version management, learners can even visualize differences between versions of a MUPPLE page and choose to go back to a previous version. Third, our prototype provides multiple views with incremental disclosure (DG 5.3) for a MUPPLE page, e.g. by switching between the different tabs. Fourth, our approach integrates the ‘development tool’ with web services (DG 5.4), which is, according to our definition, the primary goal of a mash-up personal learning environment. Fifth, sytonicity is encouraged (DG 5.5) particularly through the web-based widgets (tab ‘preview’) for user-driven development of the learning environment. Sixth, the web-based LISL interpreter allows immersion (DG 5.6) with the LISL editor at the tab ‘code’ and the command-line interpreter at the tab ‘log’. In both cases learners can play around with the LISL source code and see the effects of their modifications, this also owes to our preference for ‘designing for emergence’. Seventh, LISL supports scaffolding typical design (DG 5.7) in the way that facilitators and peers can share activity patterns which other learners can use and adapt. For that purpose we realized the before-mentioned recommendation service as a support for learners. Finally, the whole MUPPLE approach aims at community building and tools to achieve community-based activities (DG 5.8). Overall, we think that LISL and the idea of mash-up personal learning environments perfectly ties in to end-user development and that we considered relevant guidelines of this field.

4 Conclusions and Discussions

To sum up this paper, we believe that LISL and the way we designed it is a key step in realizing the three concepts of learning environment design outlined in section 1: (1) The semantic model of learning activities is the key to the ‘learning to learn’ paradigm, as we primarily focus on the learner ‘actions’ – domain-specific aspects, however, might be subject to the ‘objects’ (e.g. artifacts) available in the activities. (2) The end-user development approach itself clearly manifests that we consider the learning environment to be the outcome of and not another pre-requisite to learning. Precisely, we prefer the idea of learners developing their learning environment by themselves to instructional design or adaptive strategies. (3) Designing for emergence is important to get learners to use this way of ‘learning’ and to experience the systemic behavior as a natural part of their daily processes. With respect to the

principles for language design highlighted in this paper, we think that LISL considers them quite well, although we have not conducted evaluation studies yet.

All in all, learning environment design and LISL might be a useful answer to what Koper calls the ‘second road education’ [1] by means of lifelong learning strategies beyond primary, secondary, and tertiary education. Particularly, the MUPPLE approach is promising for building and sustaining learning networks in which actors can collaborate within shared activities and on shared objects and in which they can share best practices with peers, e.g. through LISL-based activity patterns. However, we are aware of the fact that MUPPLE is at an early development stage and, for instance, aspects of networked collaboration are still an open issue. So far, LISL aims at the learner-centered perspective, namely the learning environment design and the interaction with the tools, but lacks typical workflow issues which occur if a learner group collaborates on shared objects (artifacts) and uses the same tools. Furthermore, MUPPLE requires regulation facilities and the considerations of social network issues, like privacy, to ensure a valid learning community approach. Another weakness comprises tool interoperability, where we built upon distributed feed networks. Although we realized the FeedBack API for a few systems (Moodle, Wordpress, Scuttle), a MUPPLE-compliant learning tool must implement this API or provide some kind of generic Web 2.0 API to support interoperability according to our needs. Finally, MUPPLE lacks of evaluation results. For the design of LISL it would be necessary to show that the principles for language design, especially the user-centered ones like simplicity, learnability, efficiency, etc., are taken into account.

References

1. Koper, R.: Supporting the Continuing and Lifelong Development of Individuals in Online Learning Networks. Invited Talk at the ED-Media Conference (2008)
2. Wild, F., Mödritscher, F., Sigurdarson, S.E.: Designing for Change: Mash-Up Personal Learning Environments. In: *eLearning Papers*, 9 (2008)
3. Lieberman, H., Paterno, F., Klann, M., Wulf, V.: End-User Development: An Emerging Paradigm. In Lieberman, H., Paterno, F., Wulf, V. (eds.): *End-User Development*, LNCS, vol. 4321, pp. 1-8. Springer, Dordrecht (2006)
4. Hoare, C.A.R.: Hints on programming language design. Report, Stanford University (1973)
5. Paige, R.F., Ostroff, J.S., Brooke, P.J.: Principles for modeling language design. In: *Information and Software Technology*, 42, pp. 665-675 (2000)
6. Van Deursen, A., Klint, P., Visser, J.: Domain-specific languages: an annotated bibliography. In: *ACM SIGPLAN Notices*, 35(6), pp. 26-36 (2000)
7. Repenning, A., Ioannidou, A.: What Makes End-User Development Tick? 13 Design Guidelines. In Lieberman, H., Paterno, F., Wulf, V. (eds.): *End-User Development*, LNCS, vol. 4321, pp. 51-86. Springer, Dordrecht (2006)
8. Mödritscher, F., Neumann, G., García-Barrios, V.M., Wild, F.: A Web Application Mashup Approach for eLearning. In: *Proc. of the OpenACS/LRN Conference*, pp. 105-110 (2008)
9. Wild, F., Sigurdarson, S.E.: Distributed Feed Networks for Learning. In: *The European Journal for the Information Professional (UPGRADE)*, 9(3) (2008)

Adding mash-up based tailorability to VLEs for scripted Collaborative Learning

Juan I. Asensio-Pérez¹, Miguel L. Bote-Lorenzo¹, Guillermo Vega-Gorgojo¹,
Yannis A. Dimitriadis¹, Eduardo Gómez-Sánchez¹, Eloy D. Villasclaras-Fernández¹

¹ School of Telecom. Eng., Univ. of Valladolid, C. del Cementerio s/n, 47011 Valladolid, Spain
{juaase@tel,migbot@tel,guiveg@tel,edugom@tel,yannis@tel,evilfer@ulises.tel}.uva.es

1 Introduction

The use of Virtual Learning Environments (VLEs) is gaining momentum in education. However, most widely spread VLEs such as Moodle, Sakai or .LRN miss two important features for the support of *collaborative learning* [1] thus precluding their adoption for putting into practice this pedagogical approach that is considered to be more effective than individual and competitive learning under many circumstances. First, they are not *tailorable* [2]. In this way, educators cannot incorporate existing tools in the VLEs in order to enable their use for the support of new collaborative learning scenarios. And second, they do not support *scripting* [3]. This precludes the possibility of enhancing the effectiveness of collaborative learning by structuring the interactions between learners in the VLEs.

Gridcole [4] is a VLE proposed by the authors that overcomes these limitations by combining the IMS-LD¹ specification and service-oriented technologies. More specifically, Gridcole allows educators to easily integrate service-based tools provided by third parties for the support of scripted collaborative learning scenarios described with IMS-LD. Nevertheless, two issues have been identified that may discourage the adoption of Gridcole. One of them is that Gridcole relies on the availability of a sufficient set of tools developed by third parties and compliant with the technological requirements posed for their integration. Gridcole requires third-party tools to follow WSRF² standards that, unfortunately, are not yet commonly adopted for the development of tools with potential learning uses. The other issue refers to the fact that Gridcole was conceived to replace other VLEs. However, users (teachers, learners) may be reluctant to adopt a new VLE once they have got used to an existing VLE, despite expected benefits.

This work-in-progress paper proposes an extension for already-existing VLEs in order to add support for tailorability and scripting in collaborative learning. In this proposal, tailorability is achieved by mash-up integration [5] of third-party tools including web-based applications such as Google gadgets or Youtube-like video players. Those third-party tools also follow the principles of service-orientation advocated by Gridcole, as they provide functionality supported by third-party computational resources typically accessed through a custom-made lightweight client, but are based on technologies much more widely accepted. This opens the possibility of integrating in VLEs a large number of tools that are already available and that can

¹ IMS Learning Design. <http://www.imsglobal.org/learningdesign/>

² Web Services Resource Framework. <http://www.oasis-open.org/committees/wsrf/>

be eventually employed for the realization of many learning scenarios. In addition, the extension is aware of the activities, tools and groups defined in IMS-LD based scripts so that VLEs can rely on this information for the support of collaborative scripted learning. The aforementioned features should improve the collaborative learning support of current VLEs by enabling the integration and proper configuration of the tools required for effective interactions among learners in collaborative scenarios.

2 Proposed extension and prototype

Fig. 1 depicts the components of the proposed extension as well as the interactions that support the main use cases. Grey-filled blocks represent already existing systems (VLE, authoring tools, and third-party tools). The main element of the proposed extension is the Integration Manager that acts as an intermediary among the other blocks. As the extension intends to be valid for a wide range of existing VLEs and script authoring tools, it relies on “adaptors” that cope with the communication and data formats particularities of each of them.

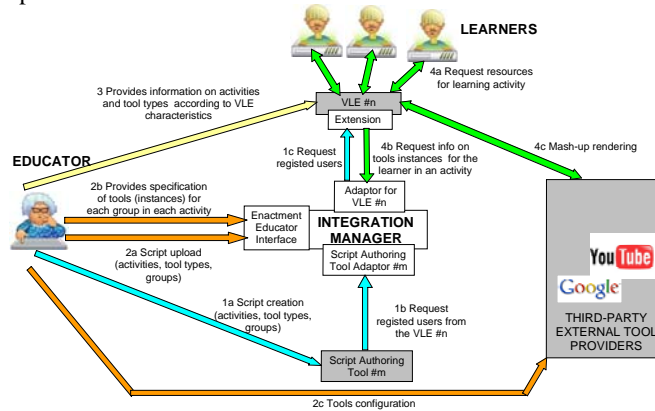


Figure 1. Blocks of the proposed extension and main interactions.

In the “Script Authoring” use case, educators describe their learning scenarios using an IMS-LD script authoring tool such as Collage³ (interaction 1a), defining the sequence of activities to be performed, the tool types required for each of them, and the groups of learners that will participate (typically different groups for different activities). In order to facilitate this authoring process, the authoring tools may request the Integration Manager to download from the VLE data on the currently registered learners (interactions 1b and 1c).

Once the script is created, the educator performs the “Mash-up Configuration” use case. Using the Enactment Educator Interface of the Integration Manager, the educator uploads the created script (interaction 2a). Next, the educator indicates the precise tool instances that will be employed by each participant or group in each activity (interaction 2b). Additionally, the educator could need to interact with the providers of particular tool instances so as to configure them (interaction 2c).

³ <http://gsic.tel.uva.es/collage>

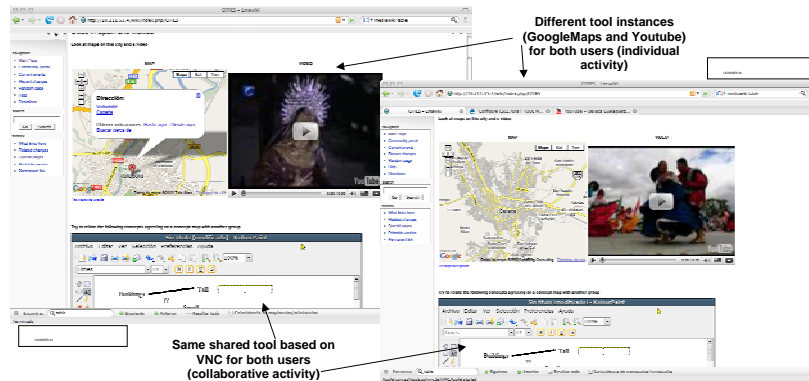


Figure 2. MediaWiki renders different mash-ups according to Integration Manager instructions.

The next use case, “VLE Configuration”, would require the educator to interact with the VLE in order to provide set-up information such as which script should be requested to the Integration Manager, how to “link” VLE pages interface with script activities or how to render subsequent tools mash-ups (interaction 3). These interactions are heavily VLE-dependant and, in some cases, they would not even be needed if the VLE adaptor for the Integration Manager is capable of carrying them out automatically.

Finally, the “Enactment” use case involves participants joining the situation through the VLE (interaction 4a) and the VLE extension asking the Integration Manager to obtain the set of tool instances required for each activity (interaction 4b). This way, tailorability is achieved through mash-up integration of external tools that actually run on the providers’ environment, while required client code for tool rendering is embedded in the VLE (interaction 4c).

A preliminary prototype has already been implemented to assess the feasibility of this proposal. MediaWiki, which was selected as the target VLE due to its simplicity, has been extended to communicate with the Integration Manager. The extension is based on MediaWiki “tag extensions” that define new wiki tags so that the educator can request the addition of desired tool types to wiki pages. The Integration Manager, implemented as a PHP module of the Drupal Content Management System, is capable of interpreting IMS-LD scripts created with the Collage authoring tool to extract the information on activities and associated tool types. Additionally, it can parse information on groups of learners created for each of the script activities provided by Collage. Furthermore, this prototype enables the integration of web-based tools such as Google gadgets or YouTube-like video players, as well as third-party standalone applications that can be shared by a group of participants via Virtual Network Computing (VNC) technology. For integrating those tools, the educator simply has to “copy and paste” the HTML code or URL indicated by the tool provider that enables the download of the tool client during the realization of the scenario.

To illustrate the functioning of this prototype, Fig. 2 shows the MediaWiki interfaces of two participants in a learning situation. Remarkably, MediaWiki shows a customized mash-up for each participant, based on the information obtained from the Integration Manager although the wiki page edited by the teacher is the same for all of them. The Integration Manager thus makes MediaWiki aware of groups, a crucial

aspect for the support of collaborative learning scenarios, and enables the educator to create a single mash-up template (in this case the edited wiki page) for all participants that is filled with appropriate tool instances (known by the Integration Manager) during the realization.

3 Discussion

VLE tailorability based on the integration of service-oriented tools is not a new concept. Nevertheless, this paper has explored a new way of achieving tailorability by using increasingly accepted mash-up techniques. This new scenario could benefit from the availability of a larger set of tools and implies an interesting shift from previous proposals (such as Gridcole) regarding the relationship among VLEs and tool providers. Using mash-ups, VLEs adapt themselves to the technology and capabilities, in terms of integration, of existing tools. No additional requirements are posed to the tool provider by the VLE (actually, there is no communication among the Integration Manager and tool providers in Fig. 1). But in some cases, this integration model could not be enough. For instance, [6] proposes the integration of third-party tools (widgets) by uploading them into an ad-hoc container that enables the easy creation by educators of tool instances according to the number of groups involved in a learning activity. However that approach poses new requirements to tool providers thus eventually resulting in a possible much smaller set of available third-party tools. Therefore, this trade-off between expected tools capabilities (more requirements on the tool-provider side) and the number of available tools (less requirements on the tool-provider side) is still an open and important issue. The authors consider that VLE tailorability should rely on different types of tool-providers (in terms of the number of requirements of the “contracts” among them and VLEs). But also that the proper choice of those types of tools needs further research on their implications on script design processes as well as on other Computer-Supported Collaborative Learning (CSCL) aspects not considered in this paper: interaction analysis, tool searching, flow of learning artifacts among tools, etc.

References

1. Dillenbourg, P.: Collaborative Learning: Cognitive and Computational Approaches. Elsevier Science, Oxford, UK (1999)
2. Morch, A.: Three Levels of End-User Tailoring: Customization, Integration and Extension. The 3rd Decennial Aarhus Conference, Aarhus, Denmark (1995) 41-45
3. Dillenbourg, P.: Over-Scripting CSCL: the Risks of Blending Collaborative Learning With Instructional Design. In: Kirschner, P. A. (eds.): Three Worlds of CSCL. Can We Support CSCL. Heerlen, Open Universiteit Nederland (2002) 61-91
4. Bote-Lorenzo, M.L., Gómez-Sánchez, E., Vega-Gorgojo, G., Dimitriadis, Y.A., Asensio-Pérez, J.I., Jorrín-Abellán, I.M.: Gridcole: A tailorable grid service based system that supports scripted collaborative learning. Computers & Education. 51 (1) (2008) 155-172
5. Severance, C., Hardin, J., Whyte, A.: The coming functionality mash-up in Personal Learning Environments. Interactive Learning Environments. 16 (1) (2008) 47-62
6. Wilson, S. Sharples, P., Griffiths, D.: Extending IMS Learning Design services using Widgets: Initial findings and proposed architecture. Open Workshop on Current research on IMS-LD and Lifelong Competence Development Infrastructures, Barcelona, Spain, 2007.

Mashups in Learning Design: pushing the flexibility envelope

*

Luis de la Fuente Valentín Derick Leony Abelardo Pardo Carlos
Delgado Kloos

Department of Telematics Engineering, Carlos III University of Madrid, Spain
<http://gradient.it.uc3m.es>

Abstract. Internet is evolving toward highly personalized user spaces. Mashups, web applications that combine data from different sources into a single tool, are providing users with higher levels of flexibility to surf the net. These new environments can be used to offer students an attractive learning environment. At a first glance, this flexibility might be contradictory with the tendency toward capturing the interaction and structure of a learning experience in languages such as IMS Learning Design. However, this paper shows an example in which these contexts can be combined to exploit both advantages. The mash-up language LISL is used to describe a flexible learning environment in the context of a Unit of Learning enacted in a learning management system.

Keywords: Design, flexibility, mash-up, service integration

1 Introduction

Learning management systems (LMS) have been evolving steadily over the last years. In the early stages, technology allowed to access most of the resources required in a learning experience such as documents, exams, quizzes, simulators, etc. The technological support in this initial stage could be defined as *information-based*. But the evolution of communication technology in general has had its effect also in the learning realm. The new scenarios are not relying only on access to information but on the interaction between all the stakeholders. Learning technology is then clearly shifting from an information-based to a *communication-based* paradigm. The tendency in the last years has shown how technology enhanced learning has been steadily adopting several interaction patterns that emerge in Internet at large.

The unfolding landscape is that of a community of users with a highly personalized environment prepared to interact with a large number of available services. The success of personalization platforms such as iGoogle, Netvibes, Pageflakes (just to mention a few) together with their potential to increase the effectiveness of a learning experiences sustains this claim. Mash-ups appear in this context and are web applications that combine data from different sources into a single tool. These sources are generically known as services.

But with the possibility of combining a set of services comes the challenge of

* Corresponding author: lfuente@it.uc3m.es

inserting such applications in current specifications used to formally capture the interaction and resources needed in a learning experience. The IMS Learning Design specification [1] (henceforth IMS LD or simply LD), provides a language capable of defining the interaction that takes place in a learning environment. While the framework claims to be generic, it is oriented toward capturing interaction at the level of the different activities within a learning experience. Furthermore, although the specification includes a property-based adaptation paradigm, these properties need to be defined at design time and therefore typically refer to aspects that are statically included in the unit of learning.

The emerging scenarios provided by the use of mash-ups needs to be combined with the description at a higher level of a Unit of Learning. The solution explored in this paper is to combine both paradigms. On one hand, the Learner Interaction Scripting Language [2] (henceforth LISL) is a language defined to create a learning environment¹ as a web-application mash-up: an aggregation of user interfaces from different tools that are combined to achieve learning outcomes. These mash-up based environments are then placed in Unit of Learning IMS LD environments, providing tools to perform course activities. The paper describes how such integration has been done in Grail [3], the Learning Design run-time environment included in the Learning Management System .LRN [4]. As a result, a Unit of Learning is used to capture the higher level organization and resource of a learning experience, but at the level of an activity, students are offered a highly flexible learning environment described in LISL.

The rest of the paper is organized as follows. Section two defines the concept of flexibility when it is applied to the enactment phase of the course life-cycle. Next, section three discuss how this type of flexibility can be offered by the joint use of IMS Learning Design and the LISL language. In section four, a flexible Unit of Learning is exposed as practical example. The article concludes with a summary of main ideas depicted in the text.

2 Flexible learning scripts

The term “script”, when applied in a pedagogical context, refers to the method to structure a learning process [5]. Course participants are guided through the flow of activities previously defined by a set of authors in a script. The level of coercion in this script will influence on the course success. Over-scripting, or a too detailed set of steps that must be followed, may reduce course effectiveness, while a too flexible scheme may not produce the expected interactions [6]. The trade-off then is to create a set of instructions detailed enough to guarantee a successful learning experience, yet leaving room for certain flexibility.

The attitude of students cannot be predicted during the authoring phase, but can be monitored while the course is being enacted specially if deployed in a LMS. Ideally, the run-time environment within the LMS should be able to react to special situations and/or minor changes in the script. This capability of changing the course behavior while is being enacted is known as *run-time flexibility* and is analyzed in detail in [7]

Typical changes in a script have a wide range of variation: structural changes are

¹ Here and hence, the term environment does not mean IMS LD environment, unless it is specifically stated by the proper use of the acronym

needed if an activity is highly dependent on previous results but they are not available at design time; content might need to be changed when participants detect anomalies in any resource; changes in group members might be needed if students drop-out. Based on these different scenarios, the flexibility required in a learning environment can be classified in different ways.

Depending on to the scope of the changes, flexibility is required at either the micro or macro script. The macro-script level refers to the high level learning flow: the activities that are present in a course, how artifacts are produced and consumed by these activities, how and when participants can be monitored and evaluated, time scheduling, or any other factor that affects the course as a whole. On the other hand, flexibility at micro-script level is related to activity-centered modifications. For example, how a single activity can be properly performed, including the tools to be used, the available content and expected interactions among peers and resources, etc.. Thus, flexibility at the micro-script level allows to re-define how an activity is performed. Typical learning scripts combine structures at both levels.

But a second type of flexibility appears depending if it is applied to teaching staff or students. Teaching staff (not necessarily authors) are in charge of supervising that the activities are taking place as expected in the script. They might need to perform changes to guarantee that the objectives can be reached. One simple example is an unexpected homework deadline extension. Students usually are not allowed to change the learning flow, but their interaction with their peers or resources can vary significantly, and therefore, the learning environment should also offer certain degree of flexibility at least to customize to certain extent the learning environment.

3 Integration of Learning Design and LISL

In computer supported learning scenarios, scripts can be formalized during the authoring phase of a course using a modelling language. All the interactions of a learning flow are then captured in a single file, which can be deployed in a compliant platform, where the enactment takes place. This section analyzes how runtime flexibility can be provided combining two different scripts: IMS Learning Design and LISL.

Macro-scripts

The IMS Learning Design specification [1] provides a framework where a wide range of pedagogies can be expressed. It is divided into different levels of complexity, each of them complements the previous one with more functionality. Thus, level A introduces the vocabulary and sets the basis of sequencing capabilities. Level B adds properties and conditions to the course definition, and allow to define several sequencing rules based on the state of the course. IMS LD is therefore a specification that matches perfectly with the concept of macro-scripts. Course structural changes during runtime can be managed by a proper use of LD properties, but they must be anticipated during the design phase. Taking advantage of this potential use of properties, LD offers an significant level of runtime flexibility [8].

From the teacher point of view, content can be made visible depending on the value of a given property, that can be modified by tutors. It is also possible to capture certain events to take place at a given time. By capturing those times in a property, modifications are easily allowed. Group management can also be captured in part by properties, allowing to dissolve and regroup students if required.

Typically, the described learning flow cannot be modified by students, but they can take decisions that affect their learning path. This is the case of profile-based statements: if students can modify their own profile, they will be able to select the kind of activity they prefer, without breaking the course constraints.

Although certain degree of flexibility can be achieved during run-time, it is at the cost of capturing these changes with numerous properties and conditions. The main consequence is that all possible changes need to be anticipated during the authoring phase. This aspect hinders significantly the possibility of performing fine-grained modifications.

Micro-scripts

The LD specification is clearly oriented to macro-script creation and therefore is not appropriate to scaffold the interaction process that occurs within an activity. A new approach is required that allows participants to select their preferred environment for the required activity. The LISL language can be used to create micro-scripts that create, manage and maintain fully-customized learning environments, and can be used to offer flexibility to students while enacting an activity [2].

The LISL language allows to define **actions** to perform in a given activity, **objects** that are produced, modified or consumed by actions, and **tool** to perform requested task. Objects can be defined by a URL, and tools must be related to a URL. LISL statements are close to natural language, so that they can be read as follows: "perform action A on object B using tool C". When the script is played, all required tools are opened in the working environment with mash-up visualization techniques.

Simplicity of LISL script creation and modification was one of the design premises of the language. As a result, the personalization of working environments becomes a really agile task that can be performed by course authors and modified by students during runtime.

Implementing the Mash-up Integration on Learning Design

In order to integrate a LISL-based mash-up with a LD player, the used mash-up templates need to be simplified to achieve a plain look. This interface simplification and the use of technologies such as AJAX helped to perform the integration very intuitively and in an unobtrusive way.

On the side of the LD player, the mash-up is referenced as a regular learning object within the environment of a given activity. Thus, the LD player will provide an independent highly customizable learning environment while remaining unaware of the level of freedom provided by the mash-up.

4 An Example

The proposed scheme to combine the two paradigms in this paper can be easily demonstrated with a simple UoL containing the previously discussed features. This section shows the deployment of a UoL as a proof of concept.

The deployment environment is based on the use of .LRN [4], a LMS built over the functionality provided by the OpenACS web toolkit. .LRN includes GRAIL as the LD run-time environment and was further modified to include Mupple, a LISL interpreter [2] and a visualization screen.

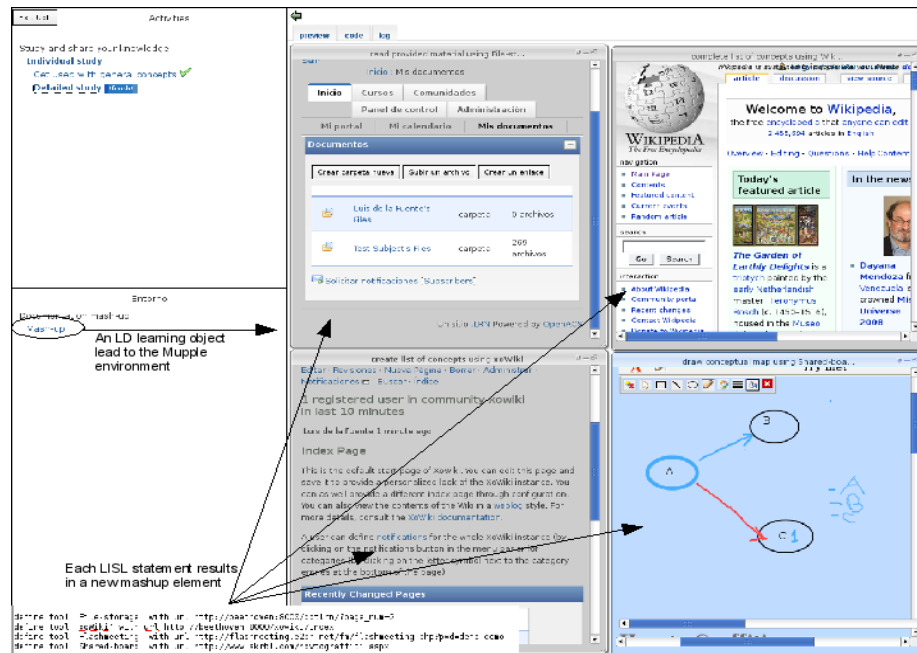


Fig. 1. Resulting working environment when combining LD and LISL.

In the UoL used in the example, the flow consists of two sessions, modeled as **acts** in LD. In the first session, learners have to study a set of initial documents and write a list of the main concepts. Then, they have to explore more deeply these concepts and draw a conceptual map linking all of them graphically. The activity starts with the reading of the documents available in the course repository. Then, students must use Wikipedia to clarify those concepts that not properly understood. Finally, a wiki tool allows to write the list of main concepts and a mind-mapper tool allows to create conceptual maps. The resulting structure is depicted in Figure 1.

Students access the activity environment and are able to modify it either modifying the source code, or using the graphical interface that allows to open, close, move and iconify each box. With this structure, an additional step (i.e. search a concept in Google), or the modification of one of the tools used by existing steps (i.e. one student prefers cmaptools for conceptual maps) can be easily achieved.

Monitor facilities are provided to allow the teacher to see what is happening in the course. This monitoring functionality is offered at the level of the LD run-time environment and it is intended to monitor at the level of macro-scripts. As it can be seen Figure 2, the platform allows the course administrator to see how many students have accessed a given activity. The same application allows also to change the termination condition of any activity or act, and to change the collection of learning objects present in an environment.

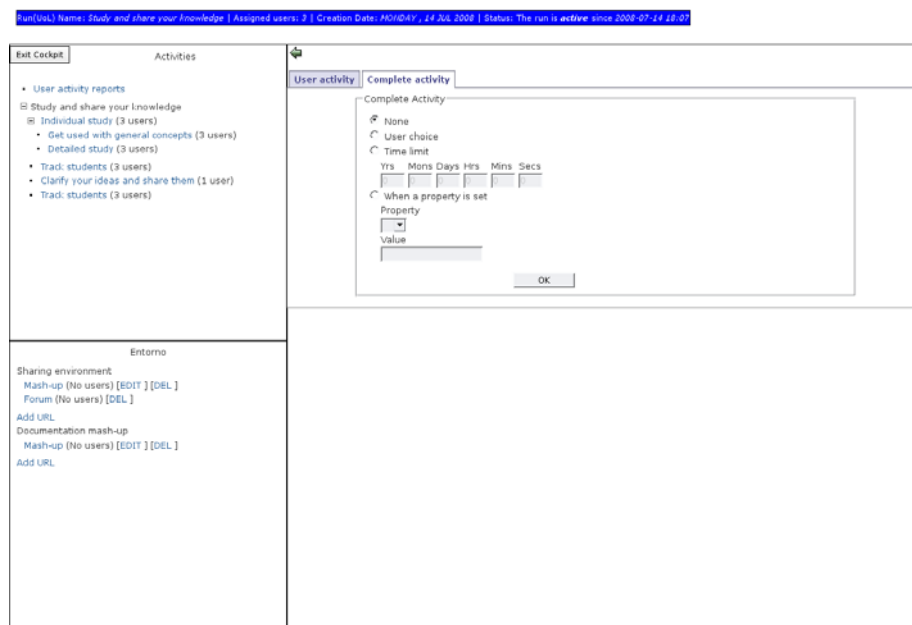


Fig. 2. Access to course details and tracking facilities.

5 Conclusions

This paper presented how flexibility on learning courses can be achieved with the joint use of LD to define the learning flow (macro-script) and LISL to detail scaffolding details on activities (micro-scripts). Flexibility is required to appear in all the course life-cycle. The proposed working scheme is focused on the run-time phase.

In this stage of the course, teachers have to be able to modify content, re-configure services, change termination conditions on an activity or its visibility. Students should have the possibility of adjusting the learning environment to their personal needs: select the tools to use and change their own profile.

The Learning Design specification offers a reasonable degree of flexibility at run-time. Level B properties and conditions can be used to modify the course behavior even if it is already running. However, this changes must be explicitly stated during the authoring phase. Used to provide a framework where activities can be performed, the LISL language allow students to easily configure the environment and share preferences with peers.

An instance of the .LRN LMS including GRAIL, the IMS LD run-time environment has been enhanced with Mupple, a LISL interpreter. This platform has been used to test this course design paradigm. An example Unit of Learning has been created and deployed, showing the potential of the architecture.

Acknowledgment. Work partially funded by *Programa Nacional de Tecnologías de la Información y de las Comunicaciones*, project TSI2005-08225-C07-01/02

References

- 1 “IMS Learning Design specification,”
http://www.imsglobal.org/learningdesign/, Feb. 2003, [On line].
- 2 F. Wild, F. Mödritscher, and S. E. Sigurdarson, “Designing for change: Mash-up personal learning environments,” *eLearning Papers*, no. 9, July 2008.
- 3 L. de la Fuente Valentín, A. Pardo, C. Delgado Kloos “Experiences with grail: Learning design support in .lrn,” in *TENCompetence Open workshop on current research in IMS Learning Design and lifelong competence development infrastructures*, June 2007.
- 4 “The .LRN platform,” http://dotlrn.org, Oct. 2007, [On line].
- 5 D. Hernández Leo, D. Burgos, C. Tattersall, and R. Koper, “Representing computer-supported collaborative learning macro-scripts using ims learning design,” in *EC-TEL (Posters)*, 2007.
- 6 P. Dillenbourg, “Over-scripting CSCL: The risks of blending collaborative learning with instructional design.” in *Three worlds of CSCL. Can we support CSCL*. Heerlen: Open University Nederland, 2002, pp. 61–91.
- 7 P. Dillenbourg and P. Tchounikine, “Flexibility in macro-scripts for computer-supported collaborative learning,” *Journal of Computer Assisted Learning*, vol. 23, no. 1, pp. 1–13, February 2007.
- 8 L. de la Fuente Valentín, A. Pardo, J. I. Asensio Pérez, Y. Dimitriadis, and C. Delgado Kloos, “Collaborative learning models on distance scenarios with learning design: a case study,” in *ICALT '08: Proc. of the eighth IEEE International Conference on Advanced Learning Technologies*. Santander, Spain: IEEE Computer Society, 2008.

Distributing education services to personal and institutional systems using Widgets

Scott Wilson, Paul Sharples, and Dai Griffiths

University of Bolton

Abstract. One of the issues for the Personal Learning Environment is the integration of personal tools with institution-centric information and services such as timetable information, and the coordination of cohort-specific group activities. One potential solution is the use of widgets – small, single-function applications that can be used in different applications, including personal technologies. One of the challenges facing the use of widgets is the diversity of platforms, and the issues around security, privacy and control. This paper describes an approach developed as part of the EU-funded TenCompetence project to develop a system based on open standards for enabling widgets to be used in a range of personal and institutional systems.

Introduction

One of the key issues in distributed e-learning is how to enable coordination to take place across a diverse range of personal as well as institutionally-managed applications in the context of formal learning. One of the key mechanisms that has been used to date has been the sharing of RSS feeds; another approach has been the use of widgets such as Facebook applications to provide institutional information in learner-managed spaces. Widgets are useful where it makes sense to offer a user interface for a service rather than a feed, or where making such a user interface available in addition to a machine-useable web API can lower the barrier to use. Examples of useful widgets within a distributed learning context include access to institutional services such as timetabling, support services, and libraries, and also widgets that enable access to shared activities with a cohort focus, such as shared chats, voting, and forums. Examples to date include the use of Facebook applications by the Open University (Hirst, 2008) and the use of OpenSocial at the University of Cambridge (Boston, 2008).

Currently, many applications provide their own plugin mechanisms to enable third-party widgets to be incorporated by users; these include learning management systems (Moodle, Blackboard), personal blogging systems (Wordpress), social networking sites (Facebook, Elgg, Ning), and the operating system itself (Apple Dashboard, Windows Sidebar). However, each system has a different API, and widgets must be developed for each one using the native programming platform of the system.

To overcome this issue a number of initiatives to standardize widget platforms have emerged. Google Gadgets and OpenSocial is one effort to create a single widget platform. The Google Gadget platform consists of a very wide range of

interconnected javascript APIs coupled with REST services to manage and deploy widgets on different social network platforms. However, the actual implementation of the platform is very complex with a lot of internal dependencies, and so an open source server solution has been created, Apache Shindig¹, to lower the barrier to entry. Although it has many good features, the Google solution remains a proprietary solution, and there has been little engagement by Google in open standards in this area.

Another initiative is the W3C's Widget specification initiative². This is an open-standards approach being developed to harmonize widgets between the Apple, Microsoft, Yahoo, Nokia and Opera platforms. While focused primarily on desktop-style widget engines, the specifications cover many of the core concerns of widget development for the web, including packaging, deployment, description, and access to APIs.

In either case, a widget is typically defined as a portable application, typically packaged in a Zip archive, and implemented using HTML, Javascript and CSS, with some deployment metadata for use by the container such as height, width, title and author information. The widget's Javascript code needs to be written to make calls to a set of standard APIs for making use of the services offered by the container. However, the two approaches differ in the package, metadata, and most importantly the API specification.

This created a dilemma for our development team, which was primarily tasked with solving issues of including tools in learning activities for the TenCompetence project³. In the end we decided to adopt the W3C approach, and extend the W3C specification to handle web-deployed widgets with collaboration features, rather than adopt the proprietary Google Gadgets platform. We created an open-source engine for these widgets and reference plugins for several container platforms.

For the future we hope to see some convergence between Google and W3C; alternatively we may work to mitigate the division between the approaches by working with the Apache Shindig project to support W3C specifications.

Architecture

The W3C model for widgets requires the use of a widget engine to provide the services for widgets such as persistence and external web access; in the case of web applications this engine needs also to communicate with the web application that is acting as the widget container. The components of the architecture we designed for our solution are shown in Figure 1.

¹ <http://incubator.apache.org/shindig/>

² <http://www.w3.org/TR/widgets/>

³ <http://www.tencompetence.org>

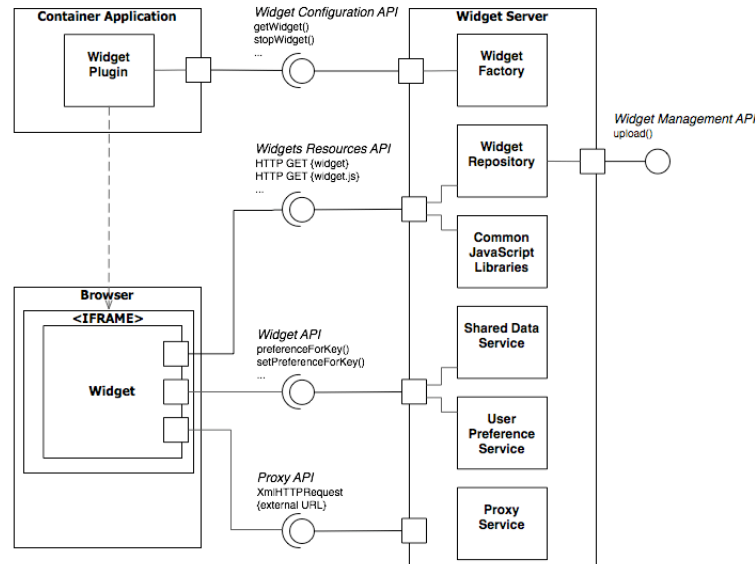


Fig. 1. Architecture of the system

Container Application

The architecture supports a wide range of potential *container applications*. Each container application must support the Widget Configuration API, and be capable of rendering a view that loads the resulting URL in a container (such as an IFrame), preferably one conforming to the dimension hints supplied in the `getWidget()` response.

The *widget plugin* enables a widget to be placed within the context of the container application, and is responsible for implementing the widget configuration API. The widget plugin collects the relevant configuration information via the container application's authoring system, and instantiates a widget using the `getWidget()` method.

Browser

Within the browser, the view rendered by the container application will typically be an IFrame that displays the widget content. The widget content is loaded from the Widget Server with the URL supplied to the plugin.

The widget is typically a small HTML file with a number of included JavaScript libraries; these include common JavaScript libraries offered by the widget server to enable widgets to communicate with the *Widget API*. Each instance of a widget (that is, a specific widget instantiated by the plugin and then displayed for a particular user) has its own unique key used in conjunction with the *Widget API* to store and retrieve

user preference information, and to access data shared across multiple instances (for example, across all chat widgets within a single course context). Widgets often require access to external services, for example to make AJAX requests for RSS feeds or to access external content; this needs to be routed via the Widget Server's *Proxy API* to avoid violating the same origin policy of the browser.

Widget Server

The Widget Server is a standalone server application that can support multiple container applications by managing and distributing widgets and offering supporting services including persistence and a URL proxy for cross-site requests.

The **Widget Factory** is responsible for instantiating and managing instances of Widgets. The Widget Factory offers the Widget Configuration API; this is used by Container Applications to instantiate widgets and to obtain configuration information including the title, URL, height and width of a widget instance.

The **Widget Repository** is responsible for managing and distributing the assets of widgets, including their HTML, CSS, images and JavaScript files. The Widget Repository offers the Widget Management API, which is used to upload and install new Widgets, and the Widget Resources API, which is simply enabling access to assets using HTTP GET.

The **Common JavaScript Libraries** are a standard set of support libraries that Widgets use to communicate with the Widget API and to enable callbacks on events. These are accessed using the Widget Resources API for access using HTTP GET.

The **Shared Data Service** persists and returns information that is shared across multiple widget instances. For example, the content of a chat conducted between several chat widget instances. The Shared Data Services is exposed using the Widget API and typically accessed via a common JavaScript library included in each widget. As well as conventional access, the Shared Data Service is configured to support Reverse AJAX, also known as Comet⁴; in this model, updates are pushed to subscribing widget instances using rapid polling. This supports widget applications such as instant messaging or online voting without the need to set up externally hosted services.

The **User Preference Service** persists and returns information persisted for a single widget instance, such as user preference settings and any other data that is unique to an individual instance of a widget. The User Preference Service is exposed using the Widget API and typically accessed via a common JavaScript library included in each widget.

The **Proxy Service** executes AJAX requests on behalf of widgets. This circumvents Same Origin Policy restrictions in the browser environment. The Proxy Service can be configured using whitelist or blacklists and should be secured for access only via widgets served by the Widget Server.

The **Widget Configuration API** is invoked by container applications to instantiate, stop, and resume widgets. The main method offered by the API is `getWidget()` which is called by the container with context information for the widget,

⁴ see e.g. [http://en.wikipedia.org/wiki/Comet_\(programming\)](http://en.wikipedia.org/wiki/Comet_(programming))

and returns a URL generated by the server for the container to render, and the height and width in pixels of the widget instance.

The container context consists of the application identifier, the context identifier (for example, a course id) and a user handle. The latter needs to be unique, but can be entirely opaque to the widget server rather than an externally-referenced user identifier. The service then uses a nonce and MD5 hashing algorithm to return an opaque widget reference number to prevent unauthorised access to widget instances.

For example, a call to `getWidget()` may return a response such as:

```
<widgetdata>

<url>http://localhost:8080/wookie/wservices/www.tencomp
etence.org/widgets/WP3/natter/chat.htm?idkey=xR8OG1IFX5
8z/YVvlz910PQVtv8.eq.&url=http://localhost:8080/woo
kie/dwr/interface/widget.js&proxy=http://localhost:
8080/wookie/proxy</url>

<height>383</height>

<width>255</width>

<maximize>false</maximize>

</widgetdata>
```

The **Widget Resources API** enables access to the repository of widget assets, and can be implemented as standard web resource access.

The **Widget API** is the service that is invoked by widgets via the interface offered by a javascript Widget object called from within the Widget javascript code. The API provides methods for accessing the user preference service and shared data service. This API is based upon the W3C Widget Services and Events specification.

The **Proxy API** provides a mechanism for widgets to request external URLs without breaking the Same Origin Policy. For example, a typical AJAX request for an external API or RSS feed needs to be routed through the widget server Proxy API to prevent potential cross-site scripting vulnerabilities.

Implementation

Wookie server

We created an implementation of the widget server, which we call *Wookie*. Wookie is implemented as a standalone Java servlet application with a MySQL backend database. The application provides all the services identified in the architecture as simple REST-style calls, and provides an administration interface for uploading, tagging and deploying widgets. It also implements a whitelist function for the Proxy

API, which the administrator can configure. For the implementation of Comet functionality we used the Direct Web Remoting (DWR) open-source Java libraries⁵.

Wordpress plugin

One container application we tested Wookie with is Wordpress. Wordpress already offers its own specific PHP-based Widget API, and it was very simple to extend this to enable Wordpress to call the Widget Configuration API and render a widget in its sidebar. This means that the widgets served by Wookie can behave in the same manner as native widgets from the viewpoint of a Wordpress user. An example is shown in Figure 2.

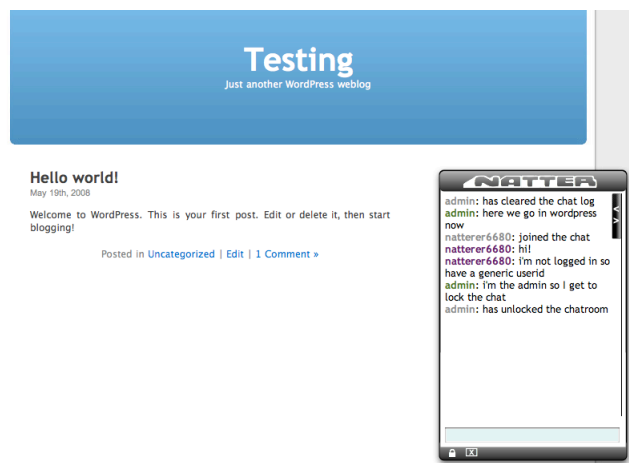


Fig. 2. Wookie widget running in Wordpress.

Moodle plugin

The second container application we developed a plugin for was the Moodle learning management system⁶. Like Wordpress, Moodle also has its own API for extensions, which Moodle calls *blocks*. The implementation of Widgets for Moodle created a new simple block type with a single configuration element for the widget type. This enables users to add Wookie widgets that then behave in the same manner as other Moodle blocks, and can be moved around the course layout as desired. An example is shown in Figure 3. Note that the “moon” and “chords” blocks are Wookie widgets

⁵ <http://directwebremoting.org/dwr/overview/dwr>

⁶ <http://www.moodle.com>

that have been converted from Apple Dashboard format to W3C format. The “natter” block is a Wookie widget we developed that uses the Shared Data Service.

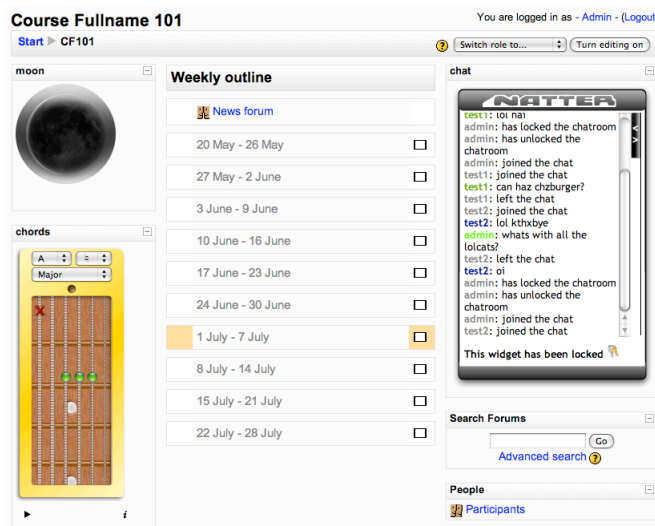


Fig. 3. Wookie widgets running in Moodle.

Collaboration Widgets

To demonstrate the capabilities of the shared data service, we created a number of collaboration widgets, including Chat, Forum, and Voting. These correspond to commonly requested features for learning designs.

Discussion

In this work we have been able to demonstrate the feasibility of extending the W3C open specification for widgets to the use of web applications, and to extend the functionality of widgets to the implementation of collaboration tools commonly employed in learning applications. By enabling such tools to be distributed in a range of containers, including personal web applications such as blogs, the potential exists to exploit this capability for offering more flexibility in the provision of e-learning.

For example, a course-cohort chat widget may be offered both through the LMS and externally through the applications in a student's own web-based PLE. We also see a strong potential role for Wookie in enabling institutional services to be embedded in both the LMS and PLE, such as timetable, support, and tutor messaging services. We see this as a pragmatic means of enabling the co-existence of PLE and LMS approaches, easing the transition from a provision-centric model to a coordination-centric model while the capabilities of the LMS and PLE converge;

without such a convergence a shift by institutions to supporting PLE users is unlikely due to the significant differences in functionality as defined by the dominant applications (Wilson et al., 2007).

Widgets also offer an easier transition for some types of institutionally managed services to be offered through personal systems; partly at least as the benefits can be expressed in terms of institutional variety management. For example, widgets can offer a relatively low-cost method for exposing services across different institutional systems such as library, MIS portal, intranet, web content management system, and LMS.

The crucial component for the future, however, will be the integration of user-centric authorization within widget architecture. The OAuth⁷ protocol offers a means for sites to establish authorization to use APIs without the transmission of user credentials. This would enable users to authorize a widget within their PLE to access data held by an institution without compromising the identity management of the institution. Currently there is considerable effort underway in the OAuth community to support widget applications, and this convergence will have a significant effect on the range of services that can be offered using widget technologies.

Future Work

For the future we intend to continue to align the architecture and implementation with the W3C specifications as they evolve towards final status, and to look into additional capabilities, such as supporting Google OpenSocial in some fashion, or integration with the Apache Shindig project, and possibly also to integrate OAuth capability. We will also continue work on supporting the use of widgets for learning designs. In the short term we intend to continue to trial the system in other contexts in order to evaluate its effectiveness, performance and usability; currently we are working with two UK-based projects involving timetable information and tutor-student messaging, and are investigating potential collaboration with several projects in other countries.

References

- Boston, Ian. (2008). Sakai and OpenSocial: A Different Approach to Distributed Learning Applications. *e-Literate*. Retrieved July 4th, 2008, from <http://www.mfeldstein.com/sakai-and-opensocial-a-different-approach-to-distributed-learning-applications/>
- Hirst, Tony. (2008). Open University Course Profiles Facebook App. *OUseful Info*. Retrieved July 4th, 2008, from <http://blogs.open.ac.uk/Maths/ajh59/010855.html>
- Wilson, S., Liber, O., Beauvoir, P., Milligan, C., Johnson, M., & Sharples, P. (2007) Personal Learning Environments: Challenging the dominant design of educational systems. **Journal of e-Learning and Knowledge Society** (2). Giunti: Genoa.

⁷ <http://oauth.net>

Increasing Widgets Interoperability at the Portal Level

Stéphane Sire

EPFL IC IIF GR-VA
BC 156 Station 14
CH-1015 Lausanne
stephane.sire@epfl.ch

Alain Vagner

CRP Henri Tudor
29, Avenue John F.Kennedy
L-1855 Luxembourg-Kirchberg
alain.vagner@tudor.lu

Abstract. We make the statement that widget APIs can be extended to create Web portals where simple interactions between widgets are possible.

Introduction

Current Web portals and portlet architectures allow user-defined placement of widgets on a multi-columns layout and the association of user preferences with each widget. This is quite convenient in a PLE to realize the visual integration of several services through the provision of a widget for each service. However, this is not enough to qualify the Portal as a mashup application, as far as the absence of communication between widgets does not allow interaction between services to be triggered from the composition portal.

In order to design a cross-widget communication mechanism to allow interaction between services at the portal level, we propose to start first with some simple scenarios. Then we discuss the requirements for a solution and we propose an extension of a widget API compatible with the Widgets 1.0 working draft of the W3C and that is being developed in the framework of the PALETTE project.

Scenarios of Cross-Widget Communication

The choice of the initiator of cross-widget communication distinguishes two types of scenarios. In the first choice, the user has the direct initiative through a drag and drop operation. In the second choice, a widget has the initiative through a change in its state, which can be triggered as a consequence a system action, or by the user.

Drag and drop Scenario

The drag and drop scenario is an extension of the drag and drop operation which is natural in direct manipulation user interfaces such as operating system desktops.

In the PALETTE project for instance, a common repository service hosts documents created by a community of practice. A search widget allows to query these documents and to present them in a result list. Some of these documents are template documents which are used in an editing service to author structured documents. In that case a convenient use of drag and drop would be to drop a template document from the search widget to the editing widget, to start the authoring of a new document from this template. In addition, the editing widget can show a preview of the template. Another example is a contact widget displaying the list of contacts of the user with eventually some online status information. Such a list is a shortcut to select a user which can then be used to perform all kinds of operation into other services. For instance, if a widget is available for negotiating meeting arrangements, it may be convenient to drag and drop the users to invite to a meeting directly from the contact widget to the meeting calendar widget.

Widget State Coupling Scenario

When a news widget receives a news item about a particular company, it could tell a stock quotes widget to display any changes in that company's share price (requirement 33 of W3C Widgets 1.0 Requirements working draft). This can be generalized to any widget that holds some data which can be updated. Updates may come from the system, or they can be triggered by user actions, such as when a user selects a buddy in her contact widget. In both cases it may be interesting to automatically propagate widget state change to any other widgets that can set their state accordingly.

Raised Issues and Requirements

The scenarios raise issues at the user level and at the technical level. At both level it is worth to search for a generic solution that would be implemented once into a portal and its widget APIs.

At the user's level, drag and drop scenarios require that users be able to detect when a drag operation is permitted from a source widget and to identify the target widgets. The first feedback is usually provided through a cursor change when flying over a drag source, and with an highlighting when flying over a potential target. This allows the provision of a generic mechanism that would work at the portal level.

State coupling scenarios require that users must be able to anticipate which widgets influence each others when they are instantiated together on the same portal. Here the designer faces with two solutions: either to let users manage explicitly the coupling between widget states, or to handle it automatically. The second case seems the easiest from a user's point of view. However, for some cases when the users wouldn't like to allow a coupling between two widgets states, it seems necessary to provide a portal with a mechanism to “decouple” two widgets, and to “undo” that decoupling. It also seems necessary to highlight the couplings that follow a widget instantiation.

At the technical level, drag and drop scenarios require a way to detect the content of a widget that can be dragged. They also require to define what data is transmitted

during the drop operation and a way to bind the transmission of data with some actions on the target widget.

State coupling scenarios require a way to describe the state of a widget and a way to bind a state update with some actions on a target widget. They also require to define a way to copy, or to access, state data of the initiating widget from bound widgets. According to the user level requirement of automatically coupling widget states, it is also required to define a protocol for matching widgets states together.

Proposition of a declarative solution

We suggest a declarative API in order to simplify the deployment of drag and drop and widget state coupling scenarios. We propose to extend the XML configuration file that comes with most widget APIs specifications, so that it can describe drag and drop bindings and state coupling bindings between widgets.

Drag and drop

The draggable content and the transmitted data are defined at the same time with a new “dragref” attribute that can be added to the presentation language (XHTML or SVG for instance) with a special namespace prefixed “w:”. Thus, the following example declares a list item which can be dragged :

```
<li w:dragref="template?http://repository.org/aFile">
  Some name</li>
```

The portal implementation uses the “dragref” attribute to generate on the fly a class attribute value that allows to define a visual alteration (such as cursor change) of the drag data island. In the future it will be possible to do it directly with CSS 3.0 namespace compatible selectors.

We propose to use a convention to declare the drag data as a combination of a keyword followed by an interrogation point followed by a URI as in the example above. The idea is that if the URI does not contain meaningful data, it can point to a Web service that would retrieve the data.

Finally, to declare bindings between some drag data and a widget as a drop target as a whole, we propose to extend the XML configuration files of the widget with a “binding” and a “submission” elements. These elements match the “dragref” contents with regular expressions. This is illustrated below. With such a declaration, the “getContact” Javascript method of the widget will be called when drag data matching the “src” attribute is dropped onto it :

```
<w:binding src="contact\?(http://id.mycop.org/.*)">
  <w:submission action="javascript:getContact"/>
</w:binding>
```

The drag data is the URI that allows to retrieve contact details from the community identity service. The Javascript method is called with the matching sub-expressions of the regular expression of the binding as parameters.

Widget state coupling

We are still investigating a syntax to declare widget state coupling in a declarative form directly within the widget configuration file.

The concepts are well defined. First, each widget must declare its state model. This could be done in a way similar to the XForms instance element. Second, each widget must declare its potential bindings with other widgets. A syntax similar to the drag and drop binding declaration above is possible, the only difference would be that instead of matching “dragref” content with regular expressions, it could be possible to use XPath expressions pointing to other widget instance element models as with XForms bindings. Finally the third step would be to describe an action to perform for each binding. Going a step further it could eventually be possible to directly link parts of the target widget instance model with corresponding parts of source widgets instance model, or to define a transformation between both, such as an XSLT one.

Related Work

Several efforts are already under way to introduce cross-widget communication into widget APIs such as into the W3C Widgets 1.0 working draft, or the Java portlet specification. However these specifications are still in development, and they do not address the user interaction issues contrary to our proposition.

The integration of services at the presentation tier is also an emerging topic, with some authors suggesting some extensions of the SOA architecture to the client side for the integration of the presentation components, while some Javascript frameworks are also appearing proposing different styles of Javascript dataflow architecture or publish-subscribe patterns to connect widgets together.

Conclusion and Future Work

The extension of Web portals toward a full Web desktop metaphor with drag and drop is intuitive. It can also be seen as a “manual” mashup, where users mix and match data coming from and going to different services with explicit actions. Actually we propose to declare the bindings into the configuration file of the widgets. This is a static solution in the sense that the bindings must be defined at the time the widget is embedded into the portal. A potentially more dynamical solution would be to define the bindings as a part of widget preferences. That would allow users to create their own bindings, or to dynamically imports new bindings as they are available.

Acknowledgements

This work is supported by the IST programme of the European Commission (DG Information Society and Media) through the PALETTE Integrated Project.

iGoogle and gadgets as a platform for integrating institutional and external services

Oskar Casquero, Javier Portillo, Ramón Ovelar, Jesús Romo, Manuel Benito

Universidad del País Vasco / Euskal Herriko Unibertsitatea (UPV/EHU)
ETSI Bilbao, Alameda Urquijo s/n, 48013 Bilbao
{oskar.casquero, javier.portillo, ramon.ovelar, jesus.romo, manuel.benito}@ehu.es

Abstract. This paper presents a framework for the integration of institutional and external services in order to give support, in a personal way, to the daily activity of each faculty member. The proposed framework is based on corporative Personal Learning Environments (corporative PLE) as the services are assembled, configured and managed within the institution. The set-up of the prototype for the development of the corporative PLE uses iGoogle and gadgets over Google Apps infrastructure. If this framework works smoothly enough, on a second phase we would like to take advantage of it as a test-bed for the research, implementation and testing of social services for educational purposes, since corporative PLE seem to be particularly effective for the creation of a network of PLE, a learning nervous system where each PLE is a neuron and which will generate some type of collective intelligence.

Keywords: corporative PLE, PLE network, social services, collective intelligence, Google Apps, iGoogle, widgets.

1 Introduction

Learning environments based on technologies that combine social services that support collaborative learning, and high personalization that supports individual characteristics and learning preferences, have the potential to radically alter the landscape of e-learning [1]. Within all learning environments, we think that the model based on a network made up of corporative Personal Learning Environments (corporative PLE) is the one that best achieves this vision [2].

This paper presents a framework based on iGoogle and gadgets over Google Apps infrastructure for the development of a network of corporative PLE. The objective is, on one hand, the integration of institutional and external services in order to give support, in a personal way, to the daily activity of each faculty member, and on the other hand, to take advantage of the framework as a test-bed for the research, implementation and testing of social services for educational purposes.

This paper is structured as follows: section 2 provides an overview of the reasons for migrating from a monolithic Virtual Learning Environment (VLE) to a network of corporative PLE; section 3 describes technological framework for a network of

corporate PLE; section 4 shows examples of the services that can be implemented on the PLE; finally, section 5 summarizes the conclusions and future work.

2 From a monolithic VLE to PLE network within the institution

The possibility for teachers to upload notes to a web page and for the students to download them is a big progress. Any of the VLE provided by institutions fulfils this function. However, our needs in learning resources, planning management and user interaction are a lot more. Moreover, from a technical point of view, although they have become more feature-rich, it is still complicated to perform upgrades and customize functionality (via APIs or otherwise). Therefore, no VLE platform will ever respond to all the needs and tastes that different teachers, students and different learning contexts will request. The VLE model makes clear that the strategy that delivers the same static learning experience to all learners and makes customization difficult does not cope with personal learning.

In recent years we have seen how social software, cloud-computing, web mashups and ubiquitous computing have changed the way we develop and use applications, and create and consume information. We can improve technology-enhanced learning if we manage to fit and guide the gradual integration of those technologies into the institutional environment [3]. In this sense, some instructors are giving the learners certain intervention grade based on Web 2.0 services. However, the amount of data generated by those services reaches such volume that they are not useful if they are not enclosed with mechanisms that enable more fluid data flow and closer user interaction. In this context, the challenge of a learner lies in the ability to find and filter out information in order to feed and keep updated user and data connections that support learning. Consequently, the need for a PLE has been identified.

The corporate PLE provides a suitable environment to improve information retrieval abilities. The deepest transformation carried out by the PLE is based on an architecture of information channels that allows to distribute any specific data among services and from a service to an interface selected by the user (web page, widget or desktop application). In order to facilitate automatized data flow, the architecture of information channels lies in the adoption of RSS syndication and open Application Programming Interfaces (API).

It is unclear the grade in which big institutions like universities will allow the use of their architecture of information channels (RSS and APIs) to facilitate the access to critical information like course enrollment data. Besides, the fact that institutions store the profile of its members gives a great opportunity to preconfigure the PLE with a set of tools, services and information channels according to such profile. Due to these reasons, it is important to select the institutional environment as the place where the tools are assembled and configured. This leads us to the idea of the corporate PLE.

Furthermore, the corporate PLE also provides a suitable environment to practice social skills. If a PLE is given to each institution member, the resulting corporate PLE network will permit learners to join into groups and deploy successfully social networks where they will perform learning experiences for many educational purposes. A corporate PLE network is a grid of learning units cooperating to share

learning resources across multiple administrative and learning contexts. This model is the one that best achieves the vision of learning as a shared nervous system, like a distributed intelligence where the knowledge emerges from collaborative processes developed by all the users.

3 Technological platform for a network of PLEs

On summer 2008, a project entitled MeMeTEKA¹ was initiated in the Faculty of Medicine at the University of the Basque Country, with the primary objective of creating a prototype of a PLE network that will show the complexities involved in the implementation of social services for educational purposes. The authors participate in MeMeTEKA with the main role of providing requirements and consultancy for the technologies that will be used within the project.

The first goal of the project has been to install and configure a prototype of the corporate PLE for a test group of instructors who will use it to develop educational materials that facilitate learning in the field of medicine during the next course. The need for rapid development of the prototype leads us to the decision of running the PLE on top of Google Apps infrastructure as it provides at no cost most of the common features needed to build the PLE: iGoogle, gadgets and Google App Engine.

iGoogle is the front-end of the corporate PLE. It provides access to a wide variety of widgets within a Locally Controlled Environment represented by Google Apps. The LCE allows the institution to preconfigure a set of fixed widgets with institutional tools, services and information channels customized according to the profile of the user, but also allows users to add their own preferred widgets.

Besides, iGoogle offers some features that seem to be very suitable for PLEs: open social, canvas view, widget sharing and automatic topic-based tab creation. Open Social defines a common API for accessing a social network's users and resources. Canvas view enables the visualization of powerful full-page widgets that can be very useful on browser-based interfaces. Widget sharing refers to the ability for sharing a widget and the data within it. With automatic topic-based tab creation you can ask iGoogle about a topic (e.g.: "learn english") and it will automatically add a tab with widgets based on the topic keywords.

Own-programmed gadgets constitute our widget platform for the integration of an increasingly number of back-ends (institutional and external services) into iGoogle. In order to do that, gadgets make use of XML, JavaScript, open APIs and REST paradigm. The open nature of these technologies allows the placement of widgets not only at iGoogle, but also at a wide range of platforms.

Google App Engine is our cloud-computing platform for the generation of server-side applications that extract and analyze the collective intelligence that emerges from the data and the interaction of many uses. These applications will implement the algorithms for building new services that detect similar elements (users or resources), recommend resources, discover groups, customize search engines, etc [4].

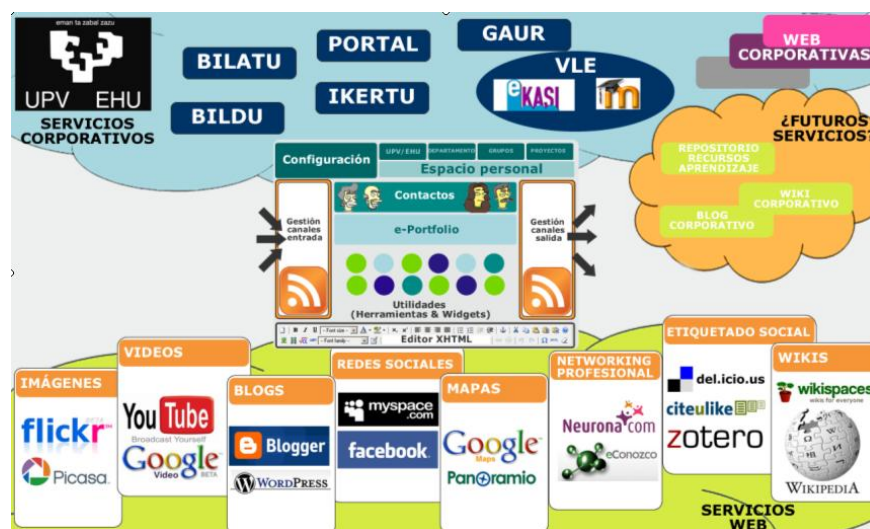
¹ <http://www.memeteka.net>

Authoring tool support is not provided by Google Apps infrastructure. Therefore, we plan to integrate eXe e-learning XHTML editor into the framework. eXe is an IMS and SCORM compliant authoring tool configured as an standalone application based on a client-server architecture. We plan to be split the server-side from the client-side. A content manager will be added to the server-side in order to distribute learning resources among the institutional and external services integrated within the framework. The client-side will be embebed as a widget and placed at iGoogle.

4 All services in one PLE

The version of iGoogle available to the users of MeMeTEKA is preconfigured with a set of fixed gadgets which include a variety of external services like Gmail, Google Calendar, Google Docs, Google Talk, Sticky Notes, Delicious, Flickr, YouTube and blogs. Gadgets for institutional services integration are not available yet as service administrators of the university have not been notified about MeMeTEKA project. When institutional services are integrated, faculty members will be linked together in groups based on grouping information (course enrollment, academic personnel and administrative personnel structure) recorded in university services.

In the near future we plan to implement several services over the network of corporative PLE. Such services include digital identity (integration with institutional LDAP services), learning resources repositories, suggestion of new widgets, creation of social networks, social graph retrieval, point to point information flow, vertical search engines and an e-portfolio to support learn-streaming.



5 Conclusions and Future work

In this paper, a framework for a network of corporative PLE has been described that, as a result of the underlying infrastructure, enables the integration of institutional and external services without requiring specialized or expensive software.

Besides, the development of a PLE for a hands-on experience in the research, implementation and testing of social services for educational purposes is one of the mayor requirements of the MeMeTEKA project and fits perfectly in the context of a PhD dissertation, in progress, about PLE and Social Networks. This approach needs an infrastructure that allows a rapid development of a prototype in order to obtain results that allow us to advance on this topic.

Google Apps offers a flexible and innovative infrastructure that is well suited for that aim. Own-programmed widgets can be used outside Google Apps infrastructure: in widget engines for Linux, Mac and Windows; in other start pages like Netvibes and, in general, in any web page, even in current VLE that support live data transport. At the time of setting up the prototype a significant problem arose when we discovered that iGoogle on Google Apps was not the latest version (for example, there is not canvas view support). Nevertheless, it is supposed that Google will update it sooner or later.

Although the solution presented in this paper leverages the proprietary software of Google, a similar framework could be also implemented using own client-server approaches or other outsourcing application service products. What changes, according to the selected solution, is the amount of support offered, and consequently, the amount of work left to do. We choose the infrastructure of Google Apps because it masks many technical issues not directly related with our research aim, and simplifies the setting up of the prototype for the project in which it may not be possible to get from the institution the needed technical staff.

References

1. Gogoulou, A., Gouli, E., Grigoriadou, M., Samarakou, M., & Chinou, D. *A Web-based Educational Setting Supporting Individualized Learning, Collaborative Learning and Assessment*. Educational Technology & Society, 10 (4), 242-256. 2007.
2. Portillo, J., Benito, M., Romo, J., Casquero, O., Ovelar, R. and Tejedor, B. *Construcción y desarrollo de Redes Sociales mediante PLEs*. Contest awards of the Social Council of the University of the Basque Country for the creation of Social Networks. January, 2008. Retrieved August 31, 2008, from <http://giel.ehu.es/02index.html>
3. Mott, J. *Open Learning Networks*. Retrieved August 31, 2008, from <http://www.jonmott.com/blog/?p=15>
4. Seragan, T. *Collective intelligence*. O'Reilly. 2008.

Social Software Modeling and Mashup based on Actors, Activities and Assets

Evgeny Bogdanov, Christophe Salzmann,
Sandy El Helou, and Denis Gillet

Ecole Polytechnique Fédérale de Lausanne (EPFL)
CH-1015, Lausanne, Switzerland
{evgeny.bogdanov,christophe.salzmann,
sandy.elhelou,denis.gillet}@epfl.ch

Abstract. *Despite the extreme diversity of Web applications, one can find similarities among them. This paper proposes an answer to the question of knowing whether it is possible to conceptually represent different Web applications in a common manner such that they can generically be integrated in other Web applications. The 3A model developed at EPFL in the framework of the European PALETTE project is used to generalize the visual and functional properties of Web applications. A Web 2.0 personal learning environment based on the 3A model called eLogbook is used as a mashup container to integrate existing Web applications. The mapping procedure is described and illustrated with the example of an instant messaging application, showing that mashup is possible with 3A model.*

Key words: web mashups, personal learning environment, social software, activity modeling

1 Introduction

In recent years the types of Web applications has increased noticeably, from social software to online auctions and collaborative environments. The various tasks and the diverse targeted public result in numerous applications that are different in both the offered functionalities and graphical design that often make them incompatible. This incompatibility triggers the question of knowing whether there is a model to describe (some of) these applications in a common way such that they can be blended in a manner that suits users needs or wills. Some advances have been made in Web design area with commonly agreed conventions for building Web applications [1]: search fields, site id or breadcrumbs navigation become common. However, these conventions only refer to the visual aspect of a Web application, not to its functional aspect. Nevertheless, similarity among Web applications at the conceptual level can also be found and mapped into a common model, which can be used for a generic representation.

This paper is organized as follows. First, Section 2 presents the 3A model and its implementation in the personal learning environment eLogbook. Then,

Section 3 depicts how the mapping of a Web application into the 3A model can be accomplished. An instant messaging mapping example and its integration in eLogbook is proposed in Section 4. Finally, Section 5 concludes and enlightens future developments.

2 eLogbook - an implementation of the 3A model

This section summarizes the 3A model description and its implementation in eLogbook that are presented in [2][3][4].

There are two well-known theories in the field of Computer Supported Co-operative Work (CSCW), activity theory [5] and distributed cognition [6]. Both theories help in learning the properties and processes of a learning system, but they do not provide concrete design specifications and cannot be directly applied for implementing a collaborative application [7]. The proposed 3A model takes its roots in activity theory, distributed cognition and actor network theory, and proposes a concrete framework for designing a collaborative web application. It consists of the three main entities Actor, Activity and Asset from which the 3A model name is derived. The main idea of the 3A model can be formulated as follows: “An Actor is producing an Asset being within an Activity”. An Actor could be a person, a software agent or any other intelligent object such as remote device. An Asset represents a document or a collection of documents or items, such as discussion thread, wiki page or image album. An Activity is the formalization of a common objective to be achieved by a group of actors. It can be the representation of a tangible space such as a classroom, or an abstract space such as a project management environment. The 3A model can be represented as a structure similar to graph. It has nodes (Actor, Activity, Asset) referred as entities connected with directed or undirected links. There can be several links between every two nodes. Each link has a specific type and weight. The richness of the proposed model also lies in the algorithms that can be used for managing and filtering the information and events related to the 3A model representation. The 3A model and its related algorithms form the core of eLogbook.

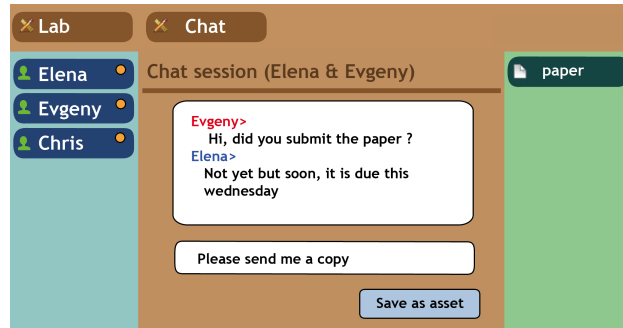


Fig. 1. Instant messaging application integrated in eLogbook framework.

eLogbook is a personal learning environment that is being developed in the framework of the European PALETTE project. To facilitate the understanding of 3A model entities and interconnections several views for eLogbook are under development. Figure 1 shows the context-specific view where the surrounding elements change dynamically in function of the central focus element to represent the relation between the central element and the other 3A entities. The order in which the surrounding elements are presented is defined by various machine learned or human suggested criteria.

3 A mashup: a web application and the 3A model

A careful examination of various Web applications (chat, wiki, forums, CMS, social network, shared repository) shows that despite their difference in goals and implementations their core features can be mapped into the 3A model (Table 1). The next section shows how this mapping is performed to provide a new functional mashup enabling the representation of differently structured Web sites in a common way. A Web mashup is a Web application that combines information from two or more external online sources [8].

Web Application	Mappings
Chat	User \rightarrow Actor Discussion \rightarrow Space Chat history \rightarrow Asset
Forum	User \rightarrow Actor Thread \rightarrow Space Post \rightarrow Asset
Social Software Facebook, MySpace	User \rightarrow Actor User profile page \rightarrow Space Uploaded file \rightarrow Asset
Shared Repository	User \rightarrow Actor Folder \rightarrow Space File \rightarrow Asset

Table 1. Web Applications mapping into 3A-model

Web applications have either explicit user accounts or implicit guest/public access. Physical users are mapped into Actors in the 3A model. Besides users, Web applications have services or other autonomous entities such as recommendation mechanisms (recommender), geo-localization mechanisms or smart devices [9] that can be represented by agents and thus mapped as Actors able to perform actions. In Web applications users either work with existing content or create and manage his/her own content. These pieces of information, such as discussions, images, documents or files, are mapped into Assets. Actors and assets are usually categorized into groups or aggregated into spaces that can be

mapped into Activities. Furthermore, interrelations among elements of a Web application are mapped into links among 3A model assets, actors and activities.

Personal Learning Environments (PLE) are systems that support the building of custom learning environment. These environments are defined and managed by learners to satisfy their learning needs. PLE can be composed of one or more components that communicate with each other. The eLogbook is such a PLE that relies on the 3A model to implement its internal functionalities.

By allowing the annotation and aggregation into different types and sources of information, eLogbook plays a key role in centralizing and contextualizing knowledge artifacts. Chat history files, to self-reflections, wikis, topics discussions in forums and external web links are all treated as assets and are centralized, annotated, and aggregated in the same way. The retrieval and exploitation of resources is done according to their labeling or tagging, the importance they were conferred through giving them a rate and the context in which they were placed. An example of contextualization is the linking of a chat discussion to a given topic within a community.

In addition, the way user interacts with Web applications is also harmonized. In fact, as it was mentioned above, the different resources are exploited, annotated and retrieved in a standard way, which moves the focus from learning how to use different applications to actually interacting with the artifacts themselves through eLogbook.

4 Integration of an instant messaging application into eLogbook

We implemented instant messaging (chat) in eLogbook in order to provide synchronous communication among users. Instant messaging application could easily be mapped into the 3A model. The proposed chat mashup is implemented into eLogbook as a helper application. Helper application represents the mapping of an original Web application into another (eLogbook) where not only the visual interface is translated but also the original functionalities. The mapping is performed as follows. First, the chat graphical interface is merged to the eLogbook as a dynamical page. Then, the chat model is mapped into the 3A model. The chat users become eLogbook actors and the discussion threads are held in spaces/activities. Often there is no mean to save chat discussions in standard applications. By using eLogbook internal functions the proposed mashup permits to effortlessly save discussions as an asset for later retrieval. The chat functionalities are triggered when the user clicks on a specific entry in the actor list or creates a new activity with a type chat. The current and the selected actors have the possibility to instantiate and perform chat conversation respectively. Once the new activity is created, the conversation can take place. The chat administrator can use the standard eLogbook means for managing activities access rights to invite or delete a user or a group of users. If an external user does not have an eLogbook account, an email address can be used. Whenever there are new messages for a given user, he/she receives a visual notification. Clicking on

the notification brings the user back to the current chat discussions. A discussion can be saved as an asset for later use by clicking on the *Save as an Asset* button (Fig. 1).

When comparing eLogbook chat implementation to the generally accepted stand-alone chat applications, the former solution shows some advantages. First, when users save discussions as assets, they can edit the saved copy to only keep the relevant parts of the discussion. The saved discussions can also be ordered (hierarchy of discussions within activities or subactivities), tagged, rated, linked and shared with other 3A model entities and/or other eLogbook members. By doing this, users contribute to building their own personal learning spaces. It should be noted, that these actions are performed within eLogbook without requiring 3rd-party applications. Awareness algorithms built in eLogbook also improve the user PLE construction by providing useful information at the adequate time.

5 Conclusions and future work

In this paper we used the 3A model to represent, at the conceptual level, Web applications and links among their entities in a common way. In many Web applications one can identify one or more of the 3A model entities (Actor, Asset and Activity). The relations among these entities can be investigated and mapped into the proposed 3A model. Once the web application structure is translated to the 3A model, the visual interface and the selected functionalities can be mapped into eLogbook to obtain a mashed up application. The integration into eLogbook allows contextualizing, centralization, and annotation of different entities or pieces of information in a similar way. The centralization offered by eLogbook improves information finding and discovering as well as information management thanks to the proposed annotation, tagging, rating and relating tools. A mashup example is illustrated by integrating an instant messaging Web application into eLogbook.

The proposed method is not yet automatic and efficient techniques for extracting the 3A-model structure from applications are to be developed.

References

1. Steve Krug. Don't Make Me Think! A Common Sense Approach to Web Usability. 2000.
2. Y. Rekik, D. Gillet, S. El Helou, and C. Salzmänn. The eLogBook Framework: Sustaining Interaction, Collaboration, and Learning in Laboratory-Oriented CoPs. *International Journal of Web-based Learning and Teaching Technologies*, 2(3):61-76, 2007.
3. D. Gillet, S. El Helou, Y. Rekik, Ch. Salzmänn: Context-Sensitive Awareness Services For Communities of Practice, 12th International Conference on Human-Computer Interaction (HCI2007), Beijing, 22-27 July (2007).
4. D. Gillet, S. El Helou, C. M. Yu, and C. Salzmänn. Turning Web 2.0 social Software into Versatile Collaborative Learning Solutions. In *The First International Conference on Advances in Computer-Human Interaction - ACHI 2008*. IEEE Computer Society Press, 2008.

5. Leont'ev, A. Problems of the development of mind. English translation, Progress Press, 1981, Moscow. (Russian original 1947).
6. Hutchins. How a Cockpit Remembers Its Speeds. Cognitive Science (1995).
7. Halverson. Activity Theory and Distributed Cognition: Or What Does CSCW Need to DO with Theories?. Computer Supported Cooperative Work (CSCW) (2002).
8. Liu, Xuanzhe, Hui, Yi, Sun, Wei and Liang, Haiqi (2007). Towards service composition based on mashup. 2007 IEEE Congress on Services, 9-13 July 2007, pp. 332-339
9. Christophe Salzmann, Denis Gillet, "From online experiments to smart devices", International Journal of Online Engineering, Vol. 4, Special Issue "REV2008", 2008, <http://www.online-journals.org/>
10. Jonassen et al. Activity theory as a framework for designing constructivist learning environments. Educational Technology Research and Development (1999).
11. Halverson. Activity Theory and Distributed Cognition: Or What Does CSCW Need to DO with Theories?. Computer Supported Cooperative Work (CSCW) (2002).
12. Erenkrantz, J. R., Gorlick, M., Suryanarayana, G., and Taylor, R. N. (2007). From representations to computations: the evolution of Web architectures. In Proceedings of the the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering. ESEC-FSE 07. ACM Press, New York, NY, p. 255-264.

A Mashup-friendly Resource and Metadata Management Framework

Hannes Ebner, Matthias Palmér

School of Computer Science and Communication
Royal Institute of Technology (KTH), Sweden
{hebner, matthias}@csc.kth.se

Abstract: Mashups and mashed up Personal Learning Environments require easy to use frameworks to support the ease of creation of effective services. The focus of this paper¹ lies on establishing a generic and mashup-friendly resource and metadata management. The assumption is that if we can find an appropriate level of generic functionality, the development of targeted tools (e.g. e-portfolios, PLEs, etc) will become a matter of user interface design and specialization. We hope that such a framework does not result in a single implementation but rather a wide variety of interoperable systems that leverage plenty of functionality. In this paper we look at already existing standards and initiatives and show why they are not sufficiently generic. We propose a framework and take recent developments into consideration. We also show an implementation and introduce a tangible use case.

Introduction

A very basic element of the Web 2.0 and Social Software is a mashup. A mashup is a (web) application that combines several data sources into one user interface or result. To make mashup creation easy, most applications provide a public API, building upon standard protocols, such as the Hyper Text Transfer Protocol (HTTP), and standard data formats, like JavaScript Object Notation (JSON) and the Extensible Markup Language (XML). A Personal Learning Environment (PLE) can be seen as a kind of a mashup. It makes the composition of a personal environment possible; built out of several (not necessarily connected) systems, tools or just data sources. Such a collection of personally aligned fragments represents the freedom of choice for learners within PLEs. A PLE does not necessarily have to be a web application, it can also exist on the desktop. It may consist of production tools (e.g. wikis and blogs), feed readers, communication and collaboration tools, social networking services, storage services, identity management, and so forth. An e-portfolio is a common component of a PLE.

On a different level, to make all this work together, some kind of resource and metadata management is needed. This means that we have to differentiate between the resource itself, its descriptive information (metadata), and administrative information such as access control, modification date, and cache control. In addition,

¹ This work has been carried out with financial support from the EU eContent^{plus} project Organic.Edunet (ECP-2006-EDU-410012), which the authors gratefully acknowledge.

we also need a differentiation between digital and non-digital resources. This approach ensures a very flexible way of managing, integrating, and reusing resources or just information about them. Splicing everything together in a simple way requires simple and powerful techniques. RESTful Web Services [1] in combination with asynchronous JavaScript and XML (AJAX) are widely used state-of-the-art technologies which allow for quick and efficient querying and modification of resources, as well as communication between services.

In order to support such a mashed up PLE infrastructure, the new version 4 of the Standardized Contextualized Access to Metadata (SCAM) framework [2] is targeted towards such environments. Instead of using an own specific data and metadata layer, applications can rely on SCAM and take advantage of its flexibility. SCAM provides a unified mechanism of accessing the managed resources and its descriptive information, which might be (re)used by any number of tools. SCAM can be seen as the least common denominator between "mashed up" applications regarding resource and metadata management.

Successively we take a look at related work, where we point to related standards and initiatives, which we discuss in the context of mashups and PLEs. Thereafter we depict a generic design of a resource and metadata management system, which also forms the basis of SCAM 4. In the following section "Implementation" we show how it is implemented, and present a use case of an application using the framework. The last section "Conclusions" reconsiders the findings during the development process and gives a perspective on applications of the framework and future developments.

Related Work

There are several standards and initiatives aiming for resource and metadata management and exchange. We briefly summarize the most important ones.

A Content Package (CP), and in particular IMS CP [3], is used to organize and package resources and describe them with metadata. The IMS CP format has been reused particularly within IMS and SCORM, for example IMS ePortfolio [5], IMS Learning Design [6], and SCORM Content Objects [4]. The standard is targeted mainly towards transfer between systems rather than providing simple access to the packaged resources. Hence, IMS CP is not optimal from a mashup perspective.

WebDAV [7] extends HTTP with functionality which allows for collaborative file management. It basically makes the WWW writable, and has support for collections, resources and links. Additional extensions enable, among other things, searching and versioning, which are important for the management of resources. Unlike HTTP, it has support for resource properties, which can be seen as limited metadata. However, reusing the same resource, describing it in different contexts, or just providing extensive metadata is not possible.

The Atom Syndication Format (Atom) [8] is based on XML and mostly used by web feeds. The complementary publishing protocol AtomPub [9] is used for creating and updating resources on the web. The basic concepts behind AtomPub are collections, workspaces, and services. A service is a grouping of workspaces,

whereas a workspace is a grouping of collections. A collection is a feed containing entries, with describing metadata for each entry. The inherent service discovery and HTTP enable a RESTful way of managing resources. There is no explicit access control except for the HTTP authentication methods, no search functionality, and no support for references, which makes it impossible to provide remote metadata. In addition and perhaps most important, creation or modification of available services, workspaces or collections is outside the scope of the protocol.

SCAM may in the end support several of these standards as a complement, however none of them do really match up for a sound architecture which supports resource and metadata management as well as interoperability and easy integration (i.e. mashups) through standards-driven design.

Discussion

The primary objective of this paper is to introduce a mechanism to manage resources and their corresponding metadata. However, the concept of resources is rather vague and we need to clarify what we mean. Resources as regular files and links to web content are commonplace. A wider perspective includes books in libraries, physical persons, calendar events, comments, concepts, and so forth. Since we aim for supporting mashups and have decided to follow the principles of REST [1], it makes sense to adhere to the definition used by the W3C Technical Architecture Group (TAG) [10] as stated in the Architecture of the World Wide Web, Volume one [11] which says:

By design a URI identifies one resource. We do not limit the scope of what might be a resource. The term "resource" is used in a general sense for whatever might be identified by a URI. It is conventional on the hypertext Web to describe Web pages, images, product catalogs, etc. as "resources". The distinguishing characteristic of these resources is that all of their essential characteristics can be conveyed in a message. We identify this set as "information resources."

This definition allows us to manage any resources that are identifiable via URIs, both "information resources" (digital resources) as well as other resources that have no digital representation. Whether a resource can be retrieved or not can be detected by trying to retrieve the resource over HTTP and inspection of the returned message. There is a recommendation by the W3C [14] on how to answer such requests. However, to follow this approach all the time is both inefficient and error prone. Servers can be down or not following the recommendation. Instead we propose that SCAM manages those pieces of information. Even if it is known that a resource is an information resource, it is unknown which format this resource is available in. This should be managed via one or several MIME types [16]. Unfortunately, the definition of resources from the W3C TAG [10] is not sufficient for our needs. For example does it not help in deciding how to distinguish a link from an uploaded file, as both can be "information resources". If a resource is managed outside the current system it should be considered to be a link. It is even possible to make a distinction

whether the metadata for the resource is managed in SCAM. Hence, we introduce the term *reference* to denote links where the metadata is managed outside of SCAM.

We introduce the concept of an entry which provides necessary information regarding the resource and the metadata for successful management in SCAM. With this definition it is more appropriate to think of a SCAM installation consisting of entries rather than of pairs of resources and metadata. Where to draw the line between what should go into the entry and what should go into the metadata is a question of pragmatism and semantics. As both the metadata and the entry will use RDF, we can build upon established standards and common practices as well the basic semantics of RDF. Providing access control on resources can be conveniently solved by expressing permissions inside the entry expression. When expressing these permissions, relevant users, groups or roles need to be available. To avoid the need for introducing additional complexity, we suggest to expose this information as specific built in resources. Other system specific entities such as ontologies, types, various configurations, etc. may also be exposed as built in resources. As these will appear as full entries with specific access control restrictions, it provides a powerful bootstrap mechanism that we envision will be used extensively.

Design

We introduce three different kinds of types that are more or less independent of each other. The *representation type* defines whether a resource has a digital representation or not. The *builtin type* indicates whether a resource gets a special treatment within SCAM. The *location type* indicates if neither, one, or both of the entry's resource and metadata is maintained within SCAM.

Representation type	Location type	Builtin type
<p><i>Information resource</i> – the resource has a representation, in the repository or elsewhere.</p> <p><i>Resolvable information resource</i> – the resource is an information resource but requires a resolvable step, e.g. through a look-up procedure that might be protocol specific such as urn:path or doi.</p> <p><i>Unknown</i> – the representation type of the resource is unknown.</p> <p><i>Named resource</i> – the resource is not an information</p>	<p><i>Local</i> – both metadata and resource are maintained in the entry's context.</p> <p><i>Link</i> – the metadata (but not the resource) is maintained in the entry's context.</p> <p><i>Link reference</i> - the resource as well as the metadata is maintained outside of the entry's context; in addition there is complementary metadata maintained in the entry's context.</p> <p><i>Reference</i> – the resource and the metadata is</p>	<p><i>Context</i> – a container resource which keeps track of a set of entries that should be managed together; at a minimum it provides default ownership of the contained entries.</p> <p><i>System context</i> – a context that is specifically treated in SCAM.</p> <p><i>Principal</i> – a user used in access control lists.</p> <p><i>Group</i> – a group used in access control lists.</p> <p><i>List</i> – an ordered list of entries.</p>

resource but it can be referred to in communication but not transferred in a message.	maintained outside the entry's context.	<i>Result list</i> – a list that is dynamically generated. <i>None</i> – all other resources without specific treatment in the repository.
---	---	---

The builtin types deserve some further consideration. Firstly, *context* is the most important of the builtin types as it divides entries into disjoint sets. In fact, every entry is required to belong to a single context. A special rule says that access to an individual entry is decided by the context's or the entry's access control list, depending on which is most permissive. Secondly, principals correspond to the users and the groups in the system, they are managed in a special context which is referred to as the *principal manager*. Thirdly, the principal manager is an example of a *system context*. Another important system context is the *context manager* which contains all *contexts* as entries. Finally, *lists* are used to organize a set of *entries* (within one context) into a ordered lists. With the help of this terminology it is much easier to introduce the RDF design, which defines how to represent entries and common metadata in RDF, and the REST design, which defines how to interact with SCAM via HTTP.

RDF Design

The context construct was introduced as SCAM context in [28] and was at that time effectively an RDF graph. With the anonymous closure algorithm, that graph was used to detect a set of statements that where managed together, referred to as a SCAM record. In this paper we introduce entries as a replacement for SCAM records. The recent introduction of Named Graphs [19] has in part invalidated the approach of SCAM contexts and records. We took the approach to use up to three Named Graphs for each entry: the administrative entry information, the entry's metadata, and sometimes even the resource itself. To achieve an aggregation of entries into a context, the context resource is a Named Graph which contains an index of all Named Graphs for all its entries. It is encouraged, but not required, to support RDF representations of entries (see the figure for a schematic picture of the RDF expression) and common metadata in the REST API. The RDF design serves the purpose as a specification language as the semantics are well defined.

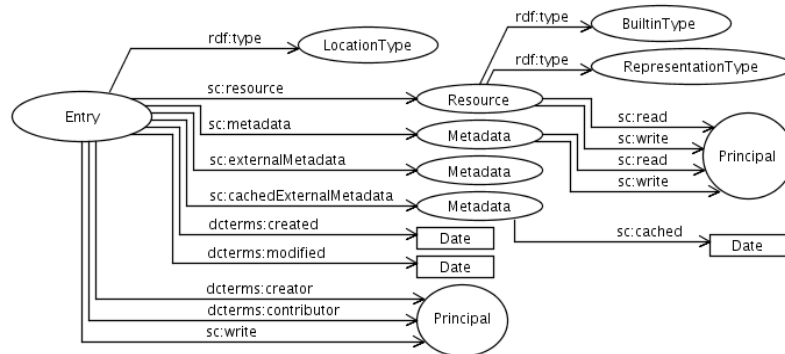


Figure: RDF design of an entry.

In contrary to the entry information, SCAM does not have an understanding of the metadata itself. This is up to the application on top of SCAM and the reason why the metadata graph cannot be generically depicted in this paper. The use case which is presented later on takes a very generic approach which allows for a flexible definition of *annotation profiles* [15] for the presentation and editing of metadata using of the SHAME library [29].

REST Design

The starting point when designing a system in a RESTful manner is to identify all things that should be accessible via separate URIs. These things are quite naturally called resources as the web is the most prime example of a REST architecture. However, as we also use the term resource in the SCAM design, this is somewhat confusing. In the following the term REST resources is used when we talk about those things that should have unique URIs in SCAM.

There are three basic kinds of REST resources in a context: resource, metadata, and entries. The following table shows the URIs and allowed HTTP operations for the three kinds of REST resources:

Operation	Method and URI
Fetch	GET {base-uri}/{context-id}/{kind}/{entry-id}
Modify	PUT {base-uri}/{context-id}/{kind}/{entry-id}
Delete	DELETE {base-uri}/{context-id}/{kind}/{entry-id}

base-uri is the base URI (namespace) that is specific for each system; *context-id* is an integer that uniquely identifies a context; *kind* is one of the three kinds of REST resources; *entry-id* is an integer that uniquely defines an entry within a context.

Resources which are links or references will most likely not have URIs that follow the pattern above and will probably only respond to GET requests. Furthermore, if the resource is not an information resource, although it is maintained locally, SCAM responds with a HTTP response containing a pointer to the URI of the entry, see the discussion in [14]. However, if it is a link or a reference, the response cannot be guaranteed as it depends on the configuration of the involved web server. For non-local metadata, i.e. when the entry indicates a reference, the URI will probably neither look like above nor will it work to fetch the metadata directly. In addition, the metadata might need to be converted from another format such as the XML binding of IEEE/LOM or extracted according to RDFa [17] or GRDDL [18].

In addition to the operations listed above, creation of new entries and listing of all entries in a context has to be possible. This is solved by introducing an additional REST resource "context". For searching we introduce a special service called "search". There are operations missing, for instance there is no way to create, delete or set access control on contexts. This is because contexts are treated as resources which are managed via entries in the context manager introduced above.

Implementation

The core of SCAM 4 is completely built on Semantic Web technologies, in particular the concept of Named Graphs [19]. With Sesame [20] we chose an open quad store with support for a variety of storage systems, high scalability, a flexible API, remote access via HTTP, several query languages, and a powerful extension API. The RESTful web services on top of the SCAM core are implemented using the Restlet framework [21], which also provides input to the upcoming Java API for RESTful Web Services (JSR-311) [22]. To serialize entries and metadata the formats RDF/XML, TriG [23], and JDIL [24] (based on JSON) are supported. In addition to RESTful web services, SCAM provides harvesting mechanisms using the protocols OAI-PMH [25] and FIRE/LRE [26]. The querying protocol SQI [27] is supported.

A Use Case: Confolio

The web-based e-portfolio Confolio is a use case where the internal types of SCAM are mapped to specific features. Contexts are used as portfolios, lists are used as folders, and entries are items in folders. In addition to the internal SCAM types, Confolio also makes use of MIME types [16]. This is necessary for the browser to know with which application a file should be opened. There are several different scenarios where Confolio can get to a meaningful deployment. As mentioned, it is appropriate as an e-portfolio, but can also be used as plain document and resource management application.

Confolio makes heavy use of the Dojo Toolkit [13]. The metadata is presented using the AJAX version of the SHAME library [29]. Confolio can be run as a stand-alone application or certain elements can be embedded into other applications.

Conclusions and Future Work

In this paper we have introduced a generic architecture and framework for resource and metadata management that is adapted to the needs of web applications, especially mashups. We believe that the most important innovation is the introduction of an entry as a solution for how to manage a resource, its metadata and the corresponding administrative information. The broad definition of an entry allows a wide range of different situations, manifested through three type schemes. These schemes are expressed in every entry and specify whether the resource is digital or not, if the resource or the metadata is managed within the repository, and if the resource is one of a few built in types that SCAM has special knowledge of. Furthermore, the architecture introduces the context resource and requires that every entry must belong to a single context. This requirement makes it possible to have an ideal administration of entries in the system by giving specific users or groups access to a single context and correspondingly to all contained entries. We provide a simple HTTP-based interface to access entries, resources and metadata according to the principles of RESTful web services. We aim to support a wide range of data formats, although we started with RDF and JSON.

The choice of HTTP in combination with the possibility of having referencing entries enables SCAM installations to be loosely connected [12], e.g., in the Confolio use case this means that folders and other resources can be unintrusively mounted across systems. Future work includes stabilizing a search API which guarantees that the access control list is respected. The plan is to have both free-text search on metadata and qualified searches using some subset of SPARQL.

References

1. Fielding, R. T.: Architectural Styles and the Design of Network-based Software Architectures, Chapter 5: Representational State Transfer (REST). Dissertation, University of California, Irvine (2000)
2. Standardized Contextualized Access to Metadata (SCAM) framework, <http://project.iml.umu.se/projects/scam>
3. IMS Content Packaging Specification, <http://www.imsglobal.org/content/packaging/>
4. Official ADL SCORM Overview, <http://www.adlnet.gov/scorm/>
5. IMS ePortfolio Specification, <http://www.imsglobal.org/ep/index.html>
6. IMS Learning Design Specification, <http://www.imsglobal.org/learningdesign/>
7. Dusseault, L. M.: HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV). RFC 4918. IETF (2007)
8. Nottingham, M., Sayre, R.: The Atom Syndication Format. RFC 4287. IETF (2005)
9. Gregorio, J., de hOra, B.: The Atom Publishing Protocol. RFC 5023. IETF (2007)
10. W3C Technical Architecture Group, <http://www.w3.org/2001/tag/>
11. Jacobs, I., Walsh, N.: Architecture of the World Wide Web, Volume One. W3C Recommendation, <http://www.w3.org/TR/webarch/> (2004)
12. Ebner, H., Palmér, M., Naeve, A.: Collaborative Construction of Artifacts, Proceedings of 4th Conference on Professional Knowledge Management, Potsdam, Germany (2007)
13. The Dojo Toolkit, <http://dojotoolkit.org>

14. Sauermann, L., Cyganiak, R.: Cool URIs for the Semantic Web. W3C Interest Group Note, <http://www.w3.org/TR/cooluris/> (2008)
15. Palmér, M., Enoksson, F., Nilsson, M., Naeve, A.: Annotation Profile Specification. Deliverable 3.2, LUISA IST-FP6-027149, <http://www.luisa-project.eu> (2007)
16. Freed, N., Borenstein, N.: Multipurpose Internet Mail Extensions (MIME), Part Two: Media Types. RFC 2046. IETF (1996)
17. Adida, B., Birbeck, M.: RDFa Primer - Bridging the Human and Data Webs. W3C Working Draft, <http://www.w3.org/TR/xhtml-rdfa-primer/> (2008)
18. Halpin, H., Davis, I.: GRDDL Primer. W3C Working Group Note, <http://www.w3.org/TR/grddl-primer/> (2007)
19. Carroll, J. J., Stickler, P.: TriX: RDF Triples in XML. Tech Report. HP Labs (2004)
20. Sesame RDF Framework, <http://openrdf.org>
21. Restlet - Lightweight REST framework for Java, <http://www.restlet.org>
22. JSR 311: JAX-RS: The Java API for RESTful Web Services, <http://jcp.org/en/jsr/detail?id=311>
23. Bizer, C., Cyganiak, R.: The TriG Syntax. <http://www4.wiwi.fu-berlin.de/bizer/TriG/Spec/>
24. JDIL - Data Integration in JSON, <http://jdil.org>
25. The Open Archives Initiative Protocol for Metadata Harvesting, <http://www.openarchives.org/OAI/openarchivesprotocol.html>
26. FIRE/LRE: The EUN Learning Resource Exchange, <http://fire.eun.org>
27. Simon, B., Massart, D., van Assche, F., Ternier, S., Duval, E.: Simple Query Interface Specification. <http://www.prolearn-project.org/lori>
28. Palmér, M., Naeve, A., Paulsson, F.: The SCAM Framework: Helping Semantic Web Applications to Store and Access Metadata. Proceedings of the European Semantic Web Symposium 2004. Springer (2004)
29. SHAME metadata library, <http://shame.sourceforge.net>

E-Learning and Microformats: A Learning Object Harvesting Model and a Sample Application

Ahmet Soylu¹, Selahattin Kuru², Fridolin Wild³, Felix Mödritscher³,

¹ K. U. Leuven, Interdisciplinary Research on Technology, Education and Communication
(iTec), Campus Kortrijk,
Etienne Sabbelaan 53, 8500 Kortrijk, Belgium
ahmet.soylu@kuleuven-kortrijk.be

² Işık University, Department of Computer Science and Engineering, Kumbaba Mevkii
Şile, 34980, Istanbul, Turkey
kuru@isikun.edu.tr

³ Vienna University of Economics and Business Administration,
Department of Information Systems, Augasse 2-6, 1090 Wien, Austria
{fridolin.wild,felix.moedritscher}@wu-wien.ac.at

Abstract. In order to support interoperability of learning tools and reusability of resources, this paper introduces a framework for harvesting learning objects from web-based content. Therefore, commonly-known web technologies are examined with respect to their suitability for harvesting embedded meta-data. Then, a lightweight application profile and a microformat for learning objects are proposed based on well-known learning object metadata standards. Additionally, we describe a web service which utilizes XSL transformation (GRDDL) to extract learning objects from different web pages, and provide a SQI target as a retrieval facility using a more complex query language called SPARQL. Finally, we outline the applicability of our framework on the basis of a search client employing the new SQI service for searching and retrieving learning objects.

Keywords: E-Learning, Learning Object, Metadata, Semantic Web, Interoperability, SQI, RDF, SPARQL, GRDDL, Microformats.

1 Motivation

Learners are not bound neither to individual learning environments as closed box of pure information nor to classical in-class learning environments anymore. Instead they are facing with several tools (web 2.0) within their learning environment, which enable them to collaborate, to reach an endless amount of information on the web, to remix and share it, and to create social networks. The e-learning market has already been overpopulated with tools and platforms to support different types of learning communities with learning management, content management, and communication facilities [1]. Depending on case, these tools are being used individually by learners, or by means of mash-ups, or as heterogeneous systems which involve several tools

and might be centered around one particular learning system. All these aspects end up in several technological challenges. Our work specially targets the following ones;

- Re-use: Learning content can be easily split up, re-used, and arranged again in order to create different contexts.
- Interoperability: Multiple learning systems or tools can communicate with each other to share and mix their learning resources.
- Accessibility: Learning content can be accessed by several learning tools and devices, besides appropriateness of content can be ensured.
- Durability: Users are not bound to any specific system, tool or device, interoperability and reusability is ensured.

We consider this work to be particularly important for mash-ups because the shift towards learner centric approaches accompanied by an ever-increasing degree of interaction and rising volumina of content provided by learners through the use of diverse social and web 2.0 tools emphasizes the importance of collective and personalized use of these distributed applications. Therefore, our work is considered as an approach to address the re-usability, interoperability, accessibility, and durability challenges of mash-ups by making existence, retrieval, sharing, and harvesting of learning objects independent of their applications, where these challenges were neglected due to the limiting features provided (RSS, import/export facilities, etc.).

2 Semantic Web

Computers are not able to automatically apply actions like ‘derive’, ‘associate’, or ‘link’ onto information available through the web – in the way humans do. Instead, they in general deal only with presentation and routine processing, as the web does most often does not provide the mark-up necessary for automatically investigating and understand the meaning of the data provided. For example, collections of documents on the web are linked to each other without allowing inspecting the motivation why these resources have been linked. The semantic web offers a variety of new opportunities to overcome these shortcomings in today’s hypertexts. Amongst others, the semantic web aims at enriching the content of the world wide web pages with meaningful structure to create an environment in which software agents can roam from page to page to carry out more intelligent tasks for their users [2]. Thereby, the semantic web integrates (earlier) research from knowledge representation, however, in the same decentralized manner the web is known for.

XML and RDF technologies constitute a basis for knowledge representation for expressing web-based semantics. An important point is that XML just describes the structure of data and does not say anything about its meaning, while RDF is used to formalize semantics. However, RDF is not enough to describe complex relationships between objects such as cardinality constraints, unions, disjoint classes etc. RDFS gives a limited capability to define only simple relationships. At this point ontologies come up as a solution, they are used to define complex relationships and sets of inference rules between data objects by providing more vocabulary. There are also many efforts for an ontology-based semantic web, important ones to mention here are

the DAML and OWL ontology languages. Both of these languages are based on XML, RDF, and RDFS. XML, RDF, and OWL can be seen as the layers where each layer requires different skills and targets different needs [3].

XML, RDF, OWL, and all of these technologies aim at expressing semantic in a machine understandable way, but not to be understood by humans. It would be a big burden to duplicate efforts by creating a piece of data in the form of both RDF, for machines, and simple XHTML, for humans. Additionally, this would also be problematic in terms of data consistency and synchronization. To avoid this double work burden, eRDF, RDFa, and microformats have been proposed, allowing for embedded semantic mark-up within web pages and being understandable for both machines and humans. RDFa is a W3C specification based on expressing structure via attributes of languages of XHTML and HTML [4]. eRDF, i.e. embedded RDF, facilitates the embedding of crucial parts of the RDF model, but does not attempt to extend this to the full RDF support [5]. eRDF is inspired by some basic principles of microformats. A microformat is a set of XHTML tags that is used to embed information into web pages which are understandable both by machines and humans while considering the human as first priority [6].

RDFa and eRDF are based on the framework provided by RDF whereas microformats offer both syntax and a restricted vocabulary that does not rely on RDF or any other framework. Therefore, microformats are domain specific. A microformat only becomes meaningful, when its syntax and vocabulary are defined by a community. This also implies that the extracting procedure is same for every eRDF and RDFa statement expressed in XHTML pages whereas it is different for every microformat. eRDF and RDFa are based on RDF. They therefore enable users to mix and use different name spaces. Microformat, however, use a flat name space that is already predefined and cannot be extended or mixed; new metadata elements cannot be added. It is obvious that RDFa and eRDF provide more flexibility and functionality than microformats, however there are hardly any real life applications of RDFa or eRDF whereas there are many real life examples of microformats. Microformats do not aim at becoming a panacea for expressing taxonomies, ontologies, or other abstractions which are not extensible or open ended [7].

While Microformats can encode explicit information to aid machine readability, they do not address implicit knowledge representation, ontological analysis, or logical inference [8]. The simplicity of this approach may be the reason of its success. Its constraints, however, are the reason why it is called 'lower case' semantic web. A living community being the key, it is not astonishing that most microformats express well-established and accepted meta-data such normalized by Dublin Core, iCal, vCard, FOAF, or ATOM. For instance, using the hAtom microformat, a website can immediately provide a feed by intelligently marking up items. There is no longer a need to publish the same content in other formats like RSS since applications can extract raw ATOM data from the (X)HTML of the website, again without converting to and from RDF [9].

Microformats are well accepted by now. Within the known limitations, they provide already simple and straightforward capabilities to manipulate meaningful structures added to human-understandable pages.

3 A Learning Object Harvesting Model for Web Content

The proposed model for harvesting learning objects bases assumes that learning objects (also called ‘learning resources’) are digital and non-digital artifacts (images, texts, etc.) that can be used for educational purposes. Learning objects can be part of the courseware ranging in size and complexity from a single graphic to an entire course [10]. Enabling learning objects to be re-usable and sharable by different tools, systems, and applications is the core challenge for e-learning interoperability.

Hereby, meta-data plays a key role: there are several bodies which have published standards and specifications to describe metadata information. In the context of this work, LOM (Learning Object Metadata) standardized by IEEE is of prominent interest, though other standards are available and compatible with the approach chosen. Most of these standards consist of many elements with varying degrees of relevance (depending mainly on the application and the usage context). For instance, LOM currently offers more than 71 elements. From this standardized set offered by LOM, an application profile has been derived mixing in elements from DC and additional custom elements. The application profile forms the basis for a microformat to embed learning resources into web pages that can later-on be harvested by crawlers. Embedding microformat structures into XHTML can be realized by hand with simple XHTML mark-up or it can be automated with the help of XSLT transformations over the XML bindings of LOM.

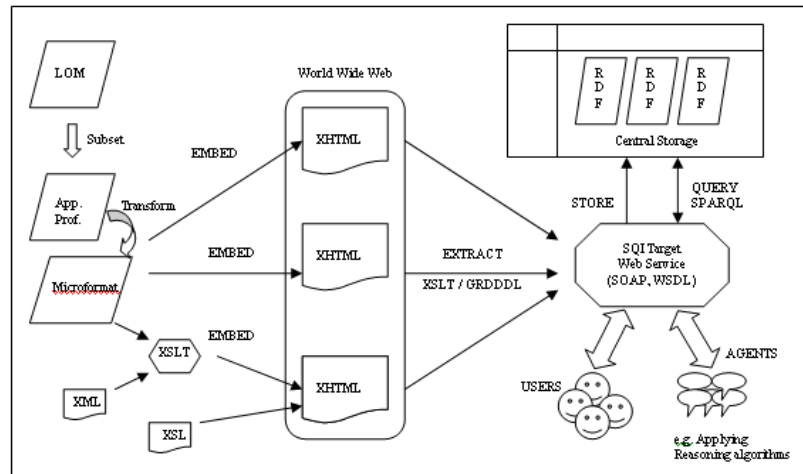


Fig. 1. Harvesting model for extracting learning object meta-data from web-based content.

After proposing a light-weight microformat for learning objects, a web service (based on SOAP and WSDL) has been created which harvests learning objects (bound to microformat) in XHTML pages and transforms them into RDF format via XSLT or XSLT and GRDDL [11] combined. If the XHTML file references an XSL file providing transformation information on how to translate microformat bindings of

learning objects into RDF then GRDDL is used. Otherwise a predefined XSL transformation is applied to perform the necessary transformation into RDF.

Additionally, an SQI target is provided which uses above mentioned harvesting mechanism and allows querying for learning objects meta-data harvested into the central storage facility. SQI is an interoperability infrastructure that enables heterogeneous systems to communicate for the purpose of learning object retrieval by using a common query language (in our case SPARQL) [12]. Fig. 1 provides an overview on the workflow of the proposed harvesting and retrieval model. As a proof of concept, a search application was implemented which uses the SQI target to query learning resource meta-data.

4 Implementation Work and Experiences

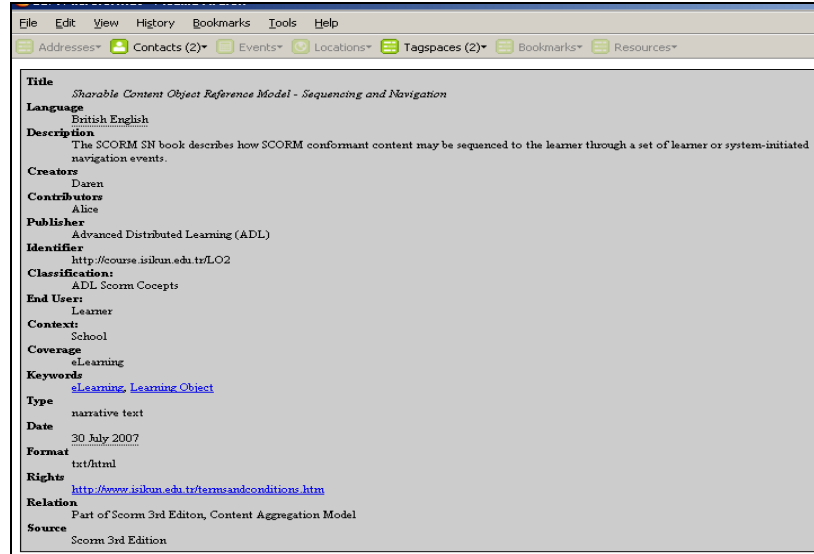
Simplicity and a minimalist approach have been considered as the guiding mantra of the prototypical implementation described below, especially while selecting data types and meta-data elements. An e-Learning application profile guideline, which was submitted to 2006 CEN/ISSS WS/LT workshop [13], has been followed to guide the construction of the application profile (roughly consists of identifying the following steps: requirement analysis, selection of data elements, multiplicity requirements, data elements from multiple name spaces, local data elements, obligation of data elements, value spaces, relationship and dependencies, and data type profiling). The primary purpose of the proposed application profile is to facilitate re-usability and interoperability of learning objects both by individuals and automated software applications such as agents. The minimalist approach defines the borders of this application profile by the vocabulary of our microformat. Therefore, it is important to address common needs instead of providing a comprehensive profile, since schemas of microformats are not subject to change often. Elements of the application profile have been derived from IEEE LOM 1482-12-1 2002 and Dublin Core, enriched by two custom elements. The application profile's reference model and XML bindings are based on SCORM 2004 3rd Edition.

The selection follows and extends the proposals for application profiles done by Friesen and Campbell [14, 15] and later work of Godby [16] (which again is based on Friesen and Campbell's spreadsheets). A total of 35 application profile has been investigated and lead to the conclusion that there is a close match between Dublin Core elements and the frequently used IEEE LOM elements. Therefore we followed the idea to map the 15 Dublin Core elements into the IEEE LOM elements if available, which means we included only LOM elements having a respective Dublin Core match, and if there is no such match, the Dublin Core element is taken as is. A total of 18 elements have been proposed, seven of them adopted from the Dublin Core name space, nine of them adopted from the IEEE LOM name space, and two additional ones that have been derived from IEEE LOM through a custom name space, this is done just in order to simplify those two aggregate elements. Tab. 1 provides an overview on the selected elements. Further details such as data types, multiplicity requirements etc. have been omitted due to space limitations.

Tab. 1. Elements of the proposed microformat (LO: LOM, DC: Dublin Core, CO: Custom)

Set	Selected Elements
LOM	Title, Language, Description, Keyword, Coverage, Type, End User, Context, Format
DC	Date, Creator, Contributor, Publisher, Rights, Relation, Source
CO	Identifier, Classification

The proposal of microformat was quite straight forward in this case, because the base applications profile itself is already simple and flat as there is no aggregate element used in application profile at all. Most of the elements has been directly associated with a ‘class’, i.e. class design pattern, attribute. The ‘creator’, ‘contributor’ and ‘publisher’ elements are used together with ‘vcard’ identifier. This is because the domains associated with these elements are based on ‘vcard’. The abbr-design pattern has been used for the ‘date’ and ‘language’ elements. The only elemental microformat, i.e. the ‘rel-tag’ microformat, is utilized for the ‘keyword’ element. Fig. 2 depicts the extraction of the proposed microformat.



Title	Sharable Content Object Reference Model - Sequencing and Navigation
Language	British English
Description	The SCORM SN book describes how SCORM conformant content may be sequenced to the learner through a set of learner or system-initiated navigation events.
Creators	Daren
Contributors	Alice
Publisher	Advanced Distributed Learning (ADL)
Identifier	http://course.isikun.edu.tr/LO2
Classification:	ADL Scorm Concepts
End User:	Learner
Context:	School
Coverage	eLearning
Keywords	eLearning , Learning Object
Type	narrative text
Date	30 July 2007
Format	txt/html
Rights	http://www.isikun.edu.tr/termsandconditions.htm
Relation	Part of Scorm 3rd Edition, Content Aggregation Model
Source	Scorm 3rd Edition

Fig. 2. Extracted meta-data of an exemplary learning object encoded with our microformat.

An SQI target has been built and enhanced with GRDDL and SPARQL by using the RAP RDF API [17]. Several sample microformat-enriched web-pages have been generated which are available to the SQI target. Then, a simple search client which is based on AJAX, JavaScript, CSS, and PHP has been set up to employ this SQI target, after its harvesting the sample pages via GRDDL, to query microformat-encoded semantics via SPARQL. These web-pages has been generated for test purposes, however any individual or authoring tool supporting our proposed microformat is capable of producing such microformat embedded pages by means of individual authoring or by means of automated transformation of current content as far as a valid

match found between their previous (or base) metadata descriptions and our microformat (this is usually the case if DC or LOM has been employed as base).

For instance the metadata for our sample learning objects are created with respect to the XML LOM bindings. Thereafter, a simple XSL transformation embeds this semantics into the web-pages according to our microformat automatically, and our SQI target harvests this semantic information using GRDDL and transforms it into RDF. Then, the query results are sent to search client in the form of a LOM XML binding, and the search client visualizes the extracted XML via XSL. In Fig. 3 the search results for the query term ‘elearning’ are displayed.

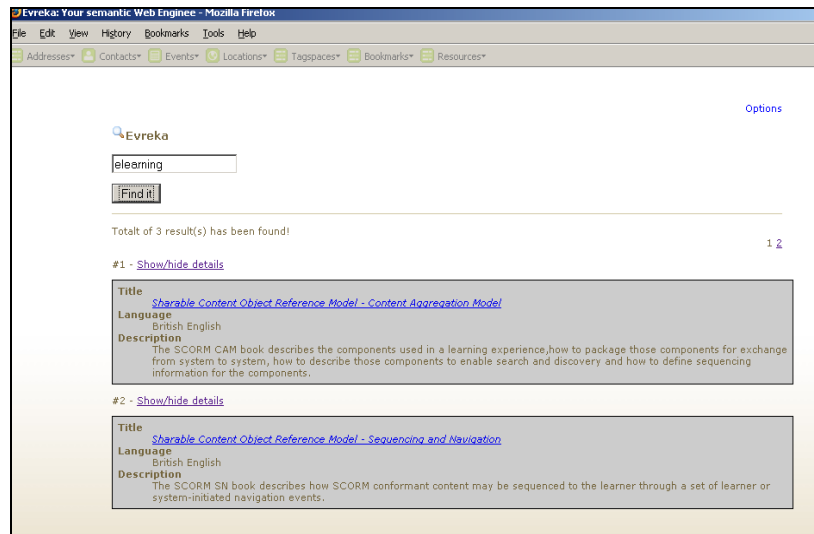


Fig. 3. Results for term ‘elearning’ visualized with our search client employing the SQI target.

It is important to highlight that the search client serves as a proof of concept, particularly for SPARQL, i.e. the SQI target only employs SPARQL’s basic facilities. Once the learning objects are represented with RDF or even empowered ontologically, query languages for RDF like SPARQL will allow the application of complex relationship queries and operations over learning objects in the sense of retrieval, analysis, content construction etc.

Comparing our microformat-based to an ontology-based approach, it has to be mentioned that microformats have clear shortcomings against RDF. Particularly, microformats lack strengths like ‘open-ended design’, ‘the ontological expression power’, and ‘extensibility’. However, a switch from Microformats to RDFa is in principle possible, though at the time of writing not promising due to the widespread acceptance and due to the ease-of-use of microformats. The drawbacks of microformats constitute the main drawbacks of the model and the sample application.

5 Conclusion and the Future Work

Within this paper, we have introduced a model and a prototypical sample application for learning object harvesting with the help of a new microformat in order to address mainly interoperability and reusability of learning content. Apart from the drawbacks which originate from the simplicity of microformats, this model needs to be complemented by strong reasoning and data mining algorithms, and, additionally, by techniques not just for harvesting objects but also for analyzing learning patterns and learning activities described on the web. Efficient agents for crawling need to be added as well. Those agents are not expected to serve just traditional desktop computers, but should rather serve any device which has access to internet such as PDAs or cell phones. In this way, learning could be embedded into real life.

References

1. Kieslinger, B., Wild, F., Arsun, O.I.: iCamp – The Educational Web for Higher Education. In: Proc. of the EC-TEL, pp. 640--645 (2006)
2. Berners-Lee, T., Hendler, J., Lassila O.: The Semantic Web. Scientific American (2001)
3. Wilson, M., Matthews, B.: The Semantic Web: Prospects and Challenges. In: Proc. Of the International Conference on Databases and Information Systems, pp. 26--29 (2006)
4. Adida, B., Birbeck, M.: RDFa Primer. W3C (2008). Available at <http://www.w3.org/TR/xhtml-rdfa-primer/#id85078> (Aug. 2008)
5. Talis: Rdf in HTML: Embedded RDF (2006). Available at <http://research.talis.com/2005/erdf/wiki/Main/RdfInHtml> (Aug. 2008)
6. Allsopp, J.: Microformats: Empowering Your Markup for Web 2.0. FriendsofED, Berkeley (2007)
7. Simpson, J.: Microformats vs. RDF: How Microformats relate to the Semantic Web (2007). Available at <http://www.semanticfocus.com/blog/entry/title/microformats-vs-rdf-how-microformats-relate-to-the-semantic-web> (Aug 2008)
8. Khare, R., Çelik, T.: Microformats: A pragmatic path to the Semantic Web. In: Proc. of the International World Wide Web Conference, pp. 865--866 (2006)
9. Graf, A.: RDFa vs. Microformats. DERI Technical Report (2007)
10. Collier, G.: e-Learning Application Infrastructure. *Whitepaper*, Sun Microsystems (2002)
11. Connolly D.: Gleaning Resource Descriptions from Dialects of Languages. W3C (2004). Available at <http://www.w3.org/2004/01/rdxh/spec> (Aug. 2008)
12. Prud'hommeaux E., Seaborne E.: SPARQL Query Language for RDF (2008). Available at <http://www.w3.org/TR/rdf-sparql-query/> (Jan. 2008)
13. CWA: Guidelines and support for building application profiles in e-learning (CEN Workshop Agreement 15555:2006E), European Committee for Standardization (2006)
14. Friesen, N.: Survey of LOM Implementations, CanCore (2003)
15. Campbell, M.L.: UK LOM Core Update. CETIS Technical Report (2003)
16. Godby, C. J.: What do Application Profiles Reveal about the Learning Object Metadata Standard? ARIADNE, October 2004 (41)
17. Melnik, S.: RAP: RDF API for PHP. SourceForge (2008). Available at <http://sourceforge.net/projects/rdfapi-php> (Aug. 2008)

Consolidating collections of learning resources using APML

Riina Vuorikari¹

¹ European Schoolnet,
Rue de Treves,
1040 Brussels, Belgium
{Riina.Vuorikari}@eun.org

Abstract. This paper explores how a group of teachers (16) create collections of digital learning resources using tagging tools. We study two different tools: an educational portal (MELT) and del.icio.us, and then propose a way to integrate the resources and tags from del.icio.us to the MELT portal using Attention Profiling Mark-up Language (APML). This allows a higher level of integration, and thus enhances the interplay among a variety of educational services.

Keywords: learning resources, tagging tools, tags, teachers.

1 Introduction

Teachers' use of digital content has been studied by [1] and [2]. To support and enhance their teaching practices, teachers acquire a variety of digital content from national and regional educational repositories (33%), use search engines on the Internet (28%), create their own content (21%) and use schoolbook publishers (7%)¹. In the context of the MELT project² we focus on teachers' access to a federation of educational repositories and to digital multilingual content.

Less attention has been paid to studying comprehensively what kind of sources of digital resources teachers use. [3] studied learning resource repositories to analyse their barriers and enablers and note that repositories are part of the repertoire of tools that individuals and communities use to achieve learning goals. Therefore, they claim, the interplay between repositories and existing tools has to be considered. We are interested in such interplay between both the official resources offered by Educational Authorities³ and what teachers find elsewhere on the Internet. The latter could be, for example, links to educational resources elsewhere on the Web or content produced by loose communities of practices who share common interests.

In our work we are interested in capturing primary and secondary school teachers' attention in using such a wide variety of digital content. Capturing and taking

¹ Replies based on 45 European primary and secondary teachers participating the MELT project

² MELT project info at <http://info.melt-project.eu>

³ More information on LRE partnerships is available at: <http://lre.eun.org>

advantage of users' interactions with the content (e.g. downloading, buying, listening, reading feeds) and users' reactions to that content (e.g. ratings, reviews, tags) is called capturing "attention data". It can be a valuable resource that reflects users' interests, activities and values, and thus serves as a proxy for their attention. Contextualized Attention Metadata (CAM) schema is built upon AttentionXML with an extension that allows capturing observations about users' activities; an example is given to trace teachers' and learners' use of the Web and digital repositories to support learning [4]. Attention Profiling Markup Language (APML⁴), on the other hand, offers a way for a user to create a personal Attention Profile, which is portable, sharable and captures users' attention related to different services.

All these specifications serve the same goal; allowing users to collect their attention and track their participation in a transparent manner while using available tools based on open standards. In this paper we focus on how we could use the existing new specifications to offer teachers a more integrated access to what they have paid attention to, i.e. their own collections of learning resources that they have created both on a digital resources portal like MELT⁵ and on del.icio.us⁶. In Section 2 we describe our case study on how teachers have used these two tagging tools to create collections of learning resources. Section 3 proposes an early idea for an integration using APML. Section 4 gives a conclusion and discusses the future work.

2 Exploratory case study on teachers' bookmarks in del.icio.us and MELT

To advance our goal to gain better understanding on teachers' usage of digital content from a variety of sources, we undertook an exploratory case study. In this initial investigation we study their collections of digital bookmarks in a real-life context of the MELT portal (*hereafter the portal*) and del.icio.us.

2.1 Method

The selected 16 teachers have both an account on the portal and on del.icio.us. They are primary and secondary teachers in science, language learning and ICTs from Finland, Estonia, Hungary and Belgium. Seven are females and nine males. One participant is under 30 years old, eight between 30-40 years, five between 40-50 years, two between 50 and 60 years old. Most of them were first introduced to del.icio.us during a summer school in 2007. In March 2008 they were invited to create a profile on the portal, where they indicated their country, interested subject areas and languages they speak. Moreover, the portal collects attention metadata regarding the learning resources bookmarked on the portal (posts). This includes information about the resource itself in LOM and the tags applied. We additionally have asked for their del.icio.us usernames to be part of this small study.

⁴ APMP, available at: <http://www.apml.org/>

⁵ MELT portal, available: <http://www.melt-project.eu>

⁶ del.icio.us, available at <http://del.icio.us>

From del.icio.us, using the html service, we were able to download these users' 100 last posts including the tags. We also recorded the total number of their posts, all the tags applied and usernames within their network. Table 1 presents the data sets; we use the term *distinct* for a tag or a resource that has been recorded in the system, as opposed to *applied*, which means how many times the tag has been associated with a post or how many times the same resource appears in collections.

Table 1: The data sets including both del.icio.us and MELT data.

	Number of users	Number of posts downloaded	Number of distinct resources	Number of distinct tags	Number of tags applied	Number of posts in the system by this group
del.icio.us	16	1176	1081	944	1583	3222
MELT	16	245	107	301	663	245

2.2 Results

Although our data sets are not directly comparable (most users have been using del.icio.us more or less for a year, whereas the portal only for 3 months), we can already see that the amount of posts in del.icio.us is substantial (3222). The median amount of posts was 105.5; 59% of subjects were above that, which indicates that most of them were dedicated del.icio.us users.

We manually analysed 50 most used distinct tags that appeared in their del.icio.us. We found that almost all of them were related to educational resources, such as teaching, English, grammar. This indicates that teachers are not using del.icio.us for personal usage, which was indicated both from the data and interviews, but are systematically collecting resources that they will use later for teaching.

3 Towards a more integrated service for teachers

We have two main aspirations for integrating teachers' del.icio.us collections in the portal. First of all, we want to make the portal a more central part of teachers' every day experience. We anticipate that if resources found via the portal and ones that they have collected from elsewhere are made available in one place, this will improve the quality of their 'digital life'. Second, currently the portal only offers resources from "trusted sources" of national educational authorities. There is no way for teachers to upload their favourite resources to the portal to share them with others (which they have requested). This deficiency could be addressed by making available both the tags and the content from teachers' del.icio.us profiles. It would also augment the number of resources that other teachers could discover.

Figure 1 displays an APML profile from del.icio.us generated by Tastebroker.org⁷. We can create a representational profile from a teacher's implicit taste based on their tags, and the explicit taste using the links that they have posted.

```
- <APML version="0.6">
- <Head>
  <Title>Del.icio.us data taste for vuorikari</Title>
  <Generator>Created by TasteBroker.org </Generator>
  <DateCreated>2008-07-11T12:22:36</DateCreated>
</Head>
- <Body defaultprofile="web">
  - <Profile name="web">
    - <ImplicitData>
      - <Concepts>
        <Concept key="ict%2C" value="0.031746034" from="tastebroker.org" updated="2008-07-11T12:22:36"/>
        <Concept key="mclt" value="0.53968257" from="tastebroker.org" updated="2008-07-11T12:22:36"/>
        <Concept key="mclt_pilot" value="0.14285715" from="tastebroker.org" updated="2008-07-11T12:22:36"/>
      </Concepts>
    </ImplicitData>
  </Profile>
</Body>
</APML>
```

Figure 1: An example of implicit taste (tags) based on an APML profile.

Experiment with APML has a number of reasons. Firstly, as we generate CAM based logs on teachers' attention on the portal, it will be in our interest to create a more holistic profile combining both del.icio.us and MELT tastes in the same APML profile. Secondly, as these profiles are portable across a number of platforms, teachers will also have a possibility to export their profile elsewhere, or import their profile from another educational service to our portal.

4 Future work

Suggestion of the future work in this area involves implementation and further development, as well as user studies on the acceptance of the idea of portability across repositories and services.

References

1. McCormick, R., Scrimshaw, P., Li, N., Carmel, C.: Celebrate Evaluation, Deliverable 7.2, European Schoolnet. (2004), Available at: http://celebrate.eun.org/eun.org2/eun/Include_to_content/celebrate/file/Deliverable7_2EvaluationReport02Dec04.pdf
2. Harley, D., Henke, J., Lawrence, S., Miller, I., Perciali, I., and Nasatir, D.: Use and Users of Digital Resources: A Focus on Undergraduate Education in the Humanities and Social Sciences. (2006) Available at: http://cshe.berkeley.edu/research/digitalresourcestudy/report/digitalresourcestudy_final_report.pdf.
3. Margaryan, A., & Littlejohn, A.: Repositories and communities at cross-purposes: Issues in sharing and reuse of digital learning resources. *Journal of Computer Assisted Learning (JCAL)*, 24(4), 333-347 (2008)
4. Najjar, J., Wolpers, M., & Duval, E. "Towards Effective Usage-Based Learning Applications: Track and Learn from User Experience(s)". (2006), In the IEEE International Conference on Advanced Learning Technologies (ICALT 2006).

⁷ Tastebrokers.org, available at: <http://tastebroker.org/>

Investigating the Suitability of Mashups for Informal Learning and Personal Knowledge Management

Sebastian Weber¹, Ludger Thomas¹, Eric Ras¹

¹ Fraunhofer IESE, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
{Forename.Surname}@iese.fraunhofer.de

Abstract. It has been shown that most of our daily learning takes place outside formal settings (e.g., seminars or trainings) where knowledge is transferred from a teacher to a learner and a certificate is often the result. This so-called informal learning is a life-long process where individuals continuously acquire knowledge. It constitutes an important element of every individual's personal knowledge management (PKM) process. Web 2.0 concepts and techniques facilitate communication with distributed individuals and help knowledge workers to cope with the immense information overload by simplifying the organization, integration, and reuse of information scattered across diverse content sources. In this paper, we investigate mashups in terms of their suitability for supporting and improving the quality of PKM and hence informal learning. This is done by analyzing which mashup features contribute to a set of information skills and learning objectives. Our analysis shows that common mashup features have the potential to broadly support designated information skills of knowledge workers.

Keywords: Mashups, Informal Learning, Personal Knowledge Management, Information Skills, Learning Objectives

1 Introduction

Knowledge workers are employees working in information-intensive organizations where they develop and use knowledge throughout their daily work. They are not supervised but manage themselves and decide on their own what and how they learn [1]. Nowadays, knowledge workers face an immense overload of information that is scattered across diverse content sources within the company (e.g., emails, file system folders, or repositories) and in the cloud (i.e., the public Web). Such information can either be pulled by the knowledge worker himself (e.g., file-system, web) or be pushed to him from an external party (e.g., the email system). Nevertheless, both approaches require effective mechanisms for finding, evaluating, structuring, and storing relevant information even if it is scattered across technical systems and organizational boundaries. This can be called Personal Knowledge Management (PKM). Once the knowledge worker has found and evaluated relevant information, he has to connect this new information to his existing knowledge structures in order to facilitate learning. From the perspective of organizational learning, he also needs to externalize,

store, and distribute his knowledge and experiences within the organization. However, knowledge management is often neglected in an organization, so that knowledge workers often perceive such activities as overhead instead of as making work easier [2].

Mashup applications seem to have the potential to broadly support PKM and to stimulate both the exchange of information and learning within an organization. The term *mashup* refers to an ad-hoc composition of information and services coming from different sources into new services [3]. Used wisely, they can help to create a corporate awareness of how employees can first refine their PKM strategies, and then contribute to the corporate knowledge. Consequently, mashups can help to establish an organizational culture of tool-supported PKM, which also comprises informal learning during daily work. In addition, Web 2.0 concepts in general and mashups in particular provide low-threshold, lightweight mechanisms for supporting the management of information and knowledge at various stages.

This paper investigates which kinds of mashups exist and how they support PKM and individual learning. For this purpose, it addresses the research question of whether specific mashups help the knowledge workers to acquire information skills as defined by Dorsey [1]. Chapter 2 defines our understanding of PKM and informal learning. Then, Chapter 3 describes common features and types of mashup applications. Chapter 4 describes the seven information skills by Dorsey [1] and maps them to the learning objectives taxonomy by Anderson and Krathwohl [12] as well as to the features of mashup applications. Chapter 5 provides a conclusion regarding the potential of PKM and informal learning mashups and suggestions for future research.

2 Informal Learning and Personal Knowledge Management

During most of their working time, knowledge workers are passing through an incremental process of acquiring, evaluating, organizing, analyzing, presenting, and securing information [1]. This is done, for example, when they go through their emails, browse the web, talk to colleagues, or read articles. Individual learning is an integral part of these PKM activities, since knowledge does not only need to be retrieved, evaluated, and managed but must somehow be transformed into some kind of personal knowledge (even if it is only knowledge about where to find the required information). The process of constructing new knowledge from an individual's information and embedding it into existing knowledge structures is called learning.

Learning is *informal* when it takes place outside formal learning settings, such as scholarly or university education where the learner usually intends to receive some kind of certificate. Such learning is an integral part of our lives – even if they were to try, humans could not “not learn”, but learning is omnipresent even outside the classroom. The Faure report (1973) [4] states that informal learning comprises about 70 percent of human learning. The notion of informal learning refers to a container term that is not clearly defined [5]. For this reason, there exist many diverse understandings about informal learning. Dohmen sees informal learning as the most fundamental form of human learning [6]. Informal learning cannot be planned or accessed, and can hardly be evaluated by looking at its ROI. This makes it difficult for organizations to

access the value of informal learning. Nevertheless, there have been several publications that show the importance of informal learning for an organization [7]. In the context of this paper, informal learning often takes place in an unconscious and accidental way (e.g., a casually learned aspect in a dialog) but can also be intended (e.g., intended learning of a topic by reading an article). Informal learning takes place outside formal settings and supports life-long learning, where the learning activities are initiated by the learner and the learning process continually adapts to the current context (e.g., the current project). A large part of informal learning happens through direct interactions with people because most knowledge is not externalized and, instead, is located inside people's minds. Mashups have the potential of fostering the willingness for and the quality of informal learning because they can bridge geographic distances, help to identify experts, provide support for interaction, pro-actively recommend context-sensitive information, or automate time-consuming processes.

In order to support informal learning, organizations should provide learning-conducive working tasks, workplaces, and working conditions. In such environments, knowledge workers can decide what they learn, how they learn (e.g., via trial and error, conventional (face-to-face) or technology-enhanced conversations (e.g., via instant messenger), examples, or observations), and from which source they will elicit knowledge. In addition, they have the time to reflect on what they learned, how they did it, and how to connect new information with existing knowledge.

Effective support of an individual's information learning activities can be seen as one contributor to the PKM process. In modern knowledge-based societies and organizations, PKM is one of the major challenges of knowledge management (KM), since it provides ways of supporting the productivity of an individual knowledge worker [2] by addressing the increasingly knowledge-intensive nature of work [8]. In addition, the goal of PKM is that employees thereby contribute to the overall corporate body of knowledge and expertise in the organization. PKM comprises managing and supporting personal knowledge and information so that it is accessible, meaningful, and valuable to the individual [2]. Thus, informal learning is an integral part. PKM also includes the organization of personal information sources in order to reduce information overload and the development of personal networks in order to foster information acquisition and informal learning. Nevertheless, the personal side of KM seems to be underestimated in many organizations [2].

3 Common Features of Mashups

Mashups as a subset of situational applications are Web-based applications that refer to an ad-hoc composition of services (either data or functionality) stemming from different sources to create entirely new services with added value that were not originally provided (and intended) by any integrated source [3, 10]. In doing so, mashup developers can leverage the whole range of Web 2.0 techniques in order to deliver rich user experience. Wong and Hong have surveyed the diverse types of mashups as well as their features [9]. The following list shows some of those types with examples:

- **Syndication:** The mashup summarizes multiple websites/services or data sets (e.g., Vidmeter.com aggregates videos charts from multiple websites, Yahoo Pipes (pipes.yahoo.com) enables users to create combined feeds).
- **Search:** The mashup provides a search over multiple external data sources (e.g., Kayak.com aggregates multiple search results from different engines).
- **Visualization:** The mashup provides the user with some kind of visualization of external data sets (e.g., liveplasma.com visually suggests to users those artists who might be of interest by leveraging the Amazon API).
- **Real-time Monitoring:** The mashup allows the user to monitor or observe one or more external services as a real-time data set (e.g., www.pimpampum.net/rt visualizes in real time the stream of information into Flickr).
- **Widget:** The mashup is a reusable micro-application that runs within a runtime environment (e.g., display of a world-time clock running on a portal or a blog).
- **Personalization:** The mashup makes use of the user's personal information exposed by other websites/services (e.g., leveraging personalized feeds of Del.icio.us as input source) or enables the construction of a personalized data set from the original service (e.g., the mobile shopping service Wishpot.com).
- **Folksonomy:** The mashup leverages a tagging mechanism for organizing its content. Tag clouds help to navigate through the content (e.g., TagBrowsr.com as an alternative way for browsing Flickr content).
- **Alternate UI & In-situ use:** A mashup provides an alternative UI to external services (e.g., oSkope.com provides a visual interface to Amazon, Ebay, etc.). In-situ use refers to mashups that support specialized usage of a service outside the typical use case (e.g., HeyWhatsThat.com utilizes GoogleMaps to describe mountains and terrain areas to users).
- **Focused View of Data:** The mashup indexes or categorizes a subset of another service's entire content (e.g., Yoututorials.com is an assemblage of video tutorials from YouTube).

In addition to this list, the following features seem useful for building mashups that support PKM. They primarily aim at social networking aspects where people are engaged in terms of commenting or rating.

- **Recommendation:** The mashup utilizes recommended content originating from social communities or where users suggest related content with regard to the content of the Web page (e.g., recommended Del.icio.us bookmarks).
- **Rating:** The mashup shows content that was either assessed by users or that has been referred to often (e.g., Amazon's stars, bookmarks in Del.icio.us).
- **Commenting/Annotating:** The mashup integrates all kinds of comments attached to some content (e.g., sticky notes in Diigo.com).
- **Sharing:** The mashup enables users to share contents with others (e.g., users share video clips within Youtube.com). Sharing requires a social networking infrastructure (e.g., Diigo.com provides friends and groups to share content with).
- **Collaboration:** The mashup enables the users to collaborate on creating or annotating content (e.g., users jointly assemble video clips in Kaltura.com).

4 Supporting Knowledge Workers through Mashups

Since Drucker [10] introduced the term knowledge worker, society in many developed countries has turned into a knowledge society (economy). Nowadays, Davenport [11] estimates that 28% of the workforce are knowledge workers who “have high degrees of expertise, education, or experience, and the primary purpose of their jobs involves the creation, distribution, or application of knowledge.”

Dorsey [1] has worked out a list of seven information skills of knowledge workers. He describes “Personal Knowledge Management, and its seven information skills, as a framework for the education of those preparing for knowledge work roles [...]” In particular, Dorsey identified:

1. **Retrieving Information** in order to “develop an understanding of the relative usefulness of these different information resources to support both [...] actions and [...] personal development” of knowledge workers
2. **Evaluating Information** in terms of evaluating and assessing information, its quality, and its relevance
3. **Organizing Information** by developing “strategies consistent with the nature of [...] work, with [...] learning styles, and [...] the nature of collaborative relationships they may have.”
4. **Collaborating around Information** means supporting “the process of working smarter, rather than merely harder, and to overcome obstacles in the form of the absence of social cues for appropriate behavior. The time spent in more face-to-face and richer electronic collaborative environments needs to be devoted to higher value activities while the actual sharing of information can be done through mechanisms that involve less collaborative activity.”
5. **Analyzing Information** “builds on the organization of information, but goes beyond it in its emphasis on the importance of frameworks, models, and theories [...]. Analysis of information addresses [...] extracting meaning out of data.”
6. **Presenting Information** “using [...] new technologies mean that increasingly knowledge workers will need to become familiar with the work of the communications specialist, the graphic designer, and the editor.”
7. **Securing Information** means that knowledge workers should frame tradeoffs regarding confidentiality, integrity, and availability.

As depicted in the table, mashups have the potential of supporting knowledge workers in acquiring a broad variety of such information skills. The middle column contains the related cognitive activities needed to achieve these skills. The verbs used in this column were defined according to the learning objective taxonomy of Anderson and Krathwohl [12]. The right column proposes assigning these cognitive abilities to practical activities in terms of mashup features (see Chapter 3). In the taxonomy, feature support from mashups ranges from basic levels such as “remembering” and “understanding” to higher levels, e.g., “analyzing” and “evaluating”.

Inform. skill	The knowledge worker does ...	Mashup features
Retrieving Information	<ul style="list-style-type: none"> • understand relationships between domain concepts. • differentiate between relevant and non-relevant key-words. ② • use the search function. • recognize already stored and evaluated sources, infor- 	<ul style="list-style-type: none"> • Folksonomy • Folksonomy ① • Search • Personalization, Focused

Inform. skill	The knowledge worker does ...	Mashup features
	mation, contacts, and knowledge based on search results. • evaluate which strategies are appropriate for retrieving relevant information. ❹	View of Data • Rating, Recommendation, Commenting/Annotating ❸
Evaluating Information	• select quality criteria and standards for evaluation and interpret the provided information. • judge the information chunks and sources based on criteria and standards.	• Personalization, Commenting/Annotating • Rating, Recommendation
Organizing Information	• represent information. • classify information chunks in such a way that they might be of future use for the learner himself or others. • construct models in order to explain semantic relationship between domain concepts. • summarize several information chunks. ❺	• Widget, Alternate UI • Folksonomy, Syndication • Folksonomy • Syndication, Focused View of Data ❻
Collaborating around Information	• collaborate by interacting, explaining, and discussing with others. • represent information to others. • compare and interpret perceptions with those of others. • share information with others (learning by teaching, e.g., by creating a tutorial).	• Collaboration • Visualization, Alternate UI, Commenting/Annotating, Widget • Rating, Commenting/Annotating • Sharing, Recommendation
Analyzing Information	• distinguish and select relevant and non-relevant information. • outline and structure chunks of information. • extract (deconstruct) common concepts.	• Commenting/Annotating, Rating, Recommendation • Commenting/Annotating • Syndication
Presenting Information	• paraphrase and represent information. • exemplify concepts and relationships. • analyze and decide whether or not to share information with others.	• Commenting/Annotating, Widget, Alternate UI • Visualization, Syndication • Rating, Commenting/Annotating, Recommendations
Securing Information	• distinguish which information is suitable to be given to outsiders and analyze how to ensure security (confidentiality) of information. ❻ • use appropriate technologies for persistent storage and sharing.	• Personalization ❼ • Sharing, Personalization

Due to the limited space in this paper, a detailed description of each entry in the table cannot be provided. Thus, in the following, we will go only into some examples indicated by numbered marks. As an example, the information skill “Retrieving Information” refers to understanding which information is needed and evaluating the appropriate strategy for getting this information (❹), e.g., by using a search function. In order to acquire this skill, differentiating between relevant and non-relevant keywords (❷) is essential. This could be supported by a folksonomy (❶), i.e., a set of commonly used tags. Tag clouds also visualize relationships between tags and indicate the relevance of each tag (i.e., keyword) by its size or color. User ratings, recommendations, and comments (❸) can be leveraged by knowledge workers for estimating the quality and relevance of the retrieved information as well as the selected retrieval strategy (e.g., whether the search engine or the keywords are bad).

Summarizing information in a knowledge worker’s own words and putting facts in a nutshell helps to increase personal learning. The underlying cognitive activity of ab-

straction and summarizing (5) is supported by the syndication features of mashups. These help knowledge workers to combine and syndicate multiple sources of information in a single place (6). In some cases, such information can be enriched and extended by the user's own annotations or documents (i.e., summaries), somewhat similar to what most people do on a paper basis. Diigo.com is an example of a social bookmarking mashup that enables users to tag and annotate websites, add them to specific topics (i.e., lists and groups), and store and share them with a group of friends. With its syndication features, Diigo.com also provides an alternate UI for consuming information from HTML pages, e.g., by extracting annotations only.

A mashup featuring personalization could allow the knowledge worker to store and retrieve user-specific information (7). Mashups that enable knowledge workers to regulate which information is public or private also ensure protection of sensitive information (8).

Finally, a few conclusive thoughts on how the other features could be leveraged by educational mashups: A mashup with an alternate UI to multiple search engines, for example, could provide relevant information and automatically link already stored, user-specific information to the search results (e.g., partially realized by social search engines). The "Focused View of Data" feature of a mashup could overlay only a specific extract of the overall personal content, namely only information corresponding to the search context.

5 Conclusion and Outlook

Although efficient PKM and informal learning have a significant impact on an employee's productivity, it is underestimated in many organizations. This paper describes how mashups can foster PKM and informal learning, in order to create and reuse personal knowledge and contribute to the overall corporate knowledge base.

Based on a literature survey as well as on personal experiences, this paper has shown that the seven information skills identified by Dorsey [1] (e.g., retrieving or analyzing information) can be mapped to features of common mashups (e.g., folksonomy). Nevertheless, empirical investigations in this field appear to be necessary for proving the validity. From our perception, at the moment there exist no explicit evaluations of Web 2.0 tools regarding the attainment of learning objectives or information skills in the literature. In this context, the work presented here serves as a baseline for making assumptions for future evaluations or even to set up a first set of formal hypotheses in the context of educational research and informal learning with mashups in particular. In the future, a more complete set of learning objectives has to be defined related to the information skills we presented here. In addition, it will be necessary to define ways for assessing what has been learned in order to measure the success of mashups for knowledge construction. Nevertheless, especially with respect to informal learning, any further investigation has to take into account that such learning "happens" and is mostly hidden to the learner himself.

However, a set of mashups that fosters the PKM process of a knowledge worker should support every information skill. Thus, knowledge workers need end-user friendly development tools for easily and quickly assembling individually tailored

“throw-away” mashups for designated contexts (e.g., current project). Existing mashup development tools aim at non-technical users and primarily address retrieving or organizing information [13] rather than helping people to gain new skills. Mashups applicable for PKM and informal or even non-formal learning should support a broader range of cognitive activities. Because those PKM mashups are individual to the highest degree and rarely usable for more than one knowledge worker, current mashup tool approaches are not sufficient, since they are specialized to a particular purpose (e.g., Yahoo Pipes combines feeds or Netvibes.com organizes information).

Instead of using single mashups, in the future it may be necessary to build some “meta mashups” (applications that are mashups of mashups) to fully support PKM. In addition to the building blocks described in chapter 3, Jay Cross presents further common web application features that can be harnessed for building meta mashups [14]. Future research has to investigate how to construct end-user friendly development tools that minimize the effort for building individual meta mashups. One research challenge would be to develop a method for how knowledge workers can be guided in constructing individual PKM environments and mashups (e.g., identifying the individual need, selecting sources and features, etc.).

References

1. Dorsey, P. A.: *Personal Knowledge Management: Educational Framework for Global Business*. Retrieved June 30, 2008, from http://www.millikin.edu/pkm/pkm_istanbul.html
2. Efimova, L.: *Understanding personal knowledge Management: A weblog case*. Enschede: Telematica Instituut (2005)
3. Duane Merrill (2006): *Mashups: The new breed of Web app*. Retrieved June 30, 2008, from <http://www.ibm.com/developerworks/library/x-mashups.html>
4. Faure, E., Herrera, F., Kaddura, A-R., Lopes, H. (1973): *Wie wir leben lernen*. Der Unesco-Bericht über Ziele und Zukunft unserer Erziehungsprogramme. Reinbek
5. Schiersmann, Ch., Remmele, H. (2002): *Neue Lernarrangements in Betrieben*. Theoretische Fundierung - Einsatzfelder - Verbreitung. In: Arbeitsgemeinschaft Betriebliche Weiterbildungsforschung e.V. (Schriften zur beruflichen Weiterbildung, Band 75). Berlin, ABWF
6. Dohmen, G. (2001): *Das informelle Lernen*. Die internationale Erschließung einer bisher vernachlässigten Grundform menschlichen Lernens für das lebenslange Lernen aller. BMBF
7. Cross, J.: *Informal Learning: Rediscovering the Natural Pathways That Inspire Innovation and Performance*, Pfeiffer (2007)
8. Drucker, P. (1999). Knowledge-worker productivity: The biggest challenge. *California Management Review*, 41, 79-94
9. Wong, J., Hong, J.: What do we "mashup" when we make mashups?. In *Proceedings of the 4th international Workshop on End-User Software Engineering* (Leipzig, Germany, May 12 - 12, 2008). WEUSE '08. ACM, New York, NY, pp. 35--39 (2008)
10. Drucker, P.: *Landmarks of tomorrow*, Harper (1959)
11. Davenport, T.: *Thinking for a Living: How to Get Better Performances And Results from Knowledge Workers*, Harvard Business School Press (2005)
12. Anderson, L.W., Krathwohl, D.R.: *A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives*. Longman, New York (2001)
13. Dion Hinchcliffe (2007): *A bumper crop of new mashup platforms*. Retrieved June 30, 2008, from <http://blogs.zdnet.com/Hinchcliffe/?p=111>

14. Jay Cross (2008): *Web 2.0 learning puzzle pieces*. Retrieved August 28, 2008, from <http://informl.com/2008/08/15/web-20-learning-puzzle-pieces/>

Maturing Learning: Mashup Personal Learning Environments

Graham Attwell^{1 2}, Jenny Bimrose², Alan Brown², Sally-Anne Barnes²

¹ Pontydysgu and ²Institute for Employment Research University of Warwick, UK
graham10@mac.com, jenny.birmose@warwick.ac.uk,
alan.brown@warwick.co.uk, sally-anne.barnes@warwick.ac.uk

Abstract. This paper provides an overview of the work of the Connexions Kent Guidance P.A.s and considers their needs in terms of knowledge maturing and development. It goes on to examine why a PLE could assist in this process and outlines the different functions of a PLE. Then a scenario is outlined illustrating the possible use of the PLE. Finally, a Mashup approach to developing the PLE is considered looking at the different possibilities for developing services for the P.A.s and developing and supporting a sustainable community of practice.

Keywords: PLEs, knowledge, learning, organisations

1. Background

This paper describes work in progress in researching and developing Personal Learning Environments (PLE) for Personal Advisors (P.A.s) employed by Connexions Kent who give labour market information (LMI) as part of careers guidance. The work is being undertaken as part of the European Seventh Framework MATURE project, which is developing technology based tools to support knowledge maturing processes. The project is based on the idea that the organisational agility has become critical for economic competitiveness. Agility requires companies to develop the competencies of their employees efficiently in order to improve the productivity of knowledge work. The failures of organisation-driven approaches to technology-enhanced learning and the success of community-driven approaches in the spirit of Web 2.0 have shown that to achieve agility, the intrinsic motivation of employees has to be harnessed so that they engage in collaborative learning activities, which can then be combined with new forms of organisational support. For that purpose, MATURE conceives individual learning processes to be linked to organisational learning in a knowledge-maturing process in which knowledge continually changes in nature. This knowledge can take many forms and be related to tasks, processes or semantic structures. The goal of MATURE is to understand this maturing process better and build tools and services to support this process including a Personal Learning & Maturing Environment (PLME), embedded into the working environment, enabling and encouraging the individual to engage in maturing activities within communities and beyond. Additionally, the project aims to develop an Organisational Learning &

Maturing Environment (OLME), enabling the organisation to analyse and to take-up community activities, to reseed innovation processes and to apply guiding strategies. The project has a number of test – or application partners – for piloting the tools and processes being developed. Connexions Kent is one of these partners.

This paper provides an overview of the work of the Connexions Kent guidance P.A.s and considers their needs in terms of knowledge maturing and development. It examines why a PLE could assist in this process and outlines the different functions of a PLE. Then, a scenario is outlined illustrating the possible use of the PLE. Finally, a Mashup approach to building a PLE is considered considering different possibilities for developing services for the P.A.s, and supporting a sustainable community of practice. It is important to note that, in this context, the use of a Mashup PLE is for knowledge development and maturing, rather than as a traditional learning application. The paper represents an initial stage in our thinking and is work in progress. At present an ethnographic study of the work of Connexions Kent P.A.s is being undertaken, which will inform further refinement of the models presented. Nevertheless, this paper will contribute to the next generation of PLE research.

2. Connexions Kent

Connexions Kent provides free impartial and confidential advice, careers guidance, support and personal development services to all 13-19 year olds, and to those up to 25 who have learning difficulties and disabilities. The service is delivered by specially trained P.A.s who are based in schools, colleges, at Connexions Access Points and in a range of community settings. Whilst the service is not restricted to careers and learning, this is a primary focus of their work. The company works with a number of partners to deliver the service, including local councils, schools and colleges, youth services, social services, health authorities, probation services, Jobcentre Plus and the police, as well as various voluntary and community groups.

Connexions Kent, like other similar organisations, are keen to engage in processes of knowledge maturing because new legislation has required fundamental changes in the structure and delivery of services for young people in England. For instance, LMI has been given a greater profile in careers guidance. These changes mean that there is a real premium on companies demonstrating how they are at the leading edge of professional developments to give them a labour market advantage for securing future contracts. Currently, and in the foreseeable future, there is a need to develop knowledge taxonomies and ontologies to support organisational development, as well as individual and community development.

Connexions Kent is keen to gain support with a process of knowledge maturing that will improve aspects of their service delivery (for example, delivering LMI more effectively as part of CEG programmes) through a bottom-up grass root expansion of an understanding of this aspect of guidance practice. The proposed collaboration with Connexions Kent means that it will offer the MATURE project a frame within which it will be possible to test the full range of knowledge maturing services, ideas and applications within a commitment to process innovation. It is expected that

continuous process improvement will be delivered through an expansion and extension of individual learning processes across the organisation.

3. Developing a PLE to support knowledge workers

The P.A.s can be considered as information knowledge workers because of the requirements: to use, routinely, different sources to obtain current and accurate information on the labour market; to extract and extrapolate key LMI; and to interpret and manipulate that data for clients of their service. Such a process involves continuous learning and knowledge development. The work of the P.A.s not only involves accessing and processing information, or explicit knowledge, but also includes the use of tacit knowledge within the organisation and the careers guidance process, ideally with tacit and explicit knowledge being combined and shared [10]. One key task for the PLE is to support these knowledge exchange processes for P.A.s. Socio-cultural theories of knowledge acquisition stress the importance of collaborative learning and 'learning communities' [7], but Agostini *et al.* [1] complain about the lack of support offered by many virtual learning environments (VLEs) for emerging communities of interest and the need to link with official organisational structures within which individuals are working. Ideally, VLEs should link knowledge assets with people, communities and informal knowledge [1] and support the development of social networks for learning [6]. The idea of a personal learning space is taken further by Razavi and Iverson [11] who suggest integrating weblogs, ePortfolios, and social networking functionality in this environment both for enhanced e-learning and knowledge management, and for developing communities of practice.

Based on these ideas of collaborative learning and social networks within communities of practice, the notion of PLEs in the workplace is being put forward as a new approach to the development of e-learning tools [2; 13] that are no longer focused on integrated learning platforms such as VLEs. In contrast, these PLEs are made-up of a collection of loosely coupled tools, including Web 2.0 technologies, used for working, learning, reflection and collaboration with others. A PLE should use social software in the workplace for informal learning which is learner driven, problem-based and motivated by interest – not as a process triggered by a single learning provider, but as a continuing activity. A development route is suggested which is based on the ideas that embedded, or work-integrated, learning support based on the pioneering ideas in the 'Learning in Process' project [12] and the APOSDLE project [8], where learning opportunities (learning objects, documents, checklists and also colleagues) are recommended based on a virtual understanding of the learner's context. While these development activities acknowledge the importance of collaboration, community engagement and of embedding learning into working processes, they have not so far addressed the linkage of individual learning processes and the further development of both individual and collective understanding as the knowledge and learning processes mature. In order to achieve that transition (to what we term a 'community of innovation'), processes of reflection and formative assessment have a critical role to play. The MATURE project will develop the idea of a PLE into a PLME by:

- § providing tools supporting the continuity of knowledge maturing from individual to community to organization (e.g. through raised awareness and consolidation tools);
- § broadening the scope of artefacts from just content, towards content, processes and semantics; and
- § connecting the tools in a meaningful way based on users' needs and embedded in linked processes of working, learning and reflection.

4. The knowledge maturing process in a PLE

Schmidt [12] and Maier & Schmidt [9] emphasise how we can gain an understanding of how knowledge maturing processes might operate within a PLE by looking at 'knowledge flows' across interlinked individual learning processes. The knowledge becomes less contextualized, more explicitly linked, easier to communicate, in short: it matures. The **maturing** of knowledge-in-practice at an individual level is represented in five phases:

- § Expressing ideas – New ideas are developed by individuals from personal experiences or in informal discussions. The knowledge is subjective and deeply embedded in the context of the originator. The vocabulary used for communication, or in private notes, is vague and often restricted to the person expressing the idea.
- § Distributing in communities – This phase accomplishes an important maturing step; the development of common terminology shared among community members (in, for example, discussion forum entries, Blog postings or wikis).
- § Formalising – Artefacts created in the preceding two phases are inherently unstructured, still highly subjective and embedded in the community context. Purpose-driven structured documents are created (e.g. reports about practice or process models in which knowledge is de-subjectified) and the context is explicit.
- § Ad-hoc learning – Documents produced in the preceding phase are not well suited as learning materials because no pedagogic considerations were taken into account. Now the topic is refined to improve comprehensibility in order to ease its use or re-use. The material is ideally prepared in a pedagogic way, enabling broader dissemination by linking general learning objectives with case examples.
- § Formal training – The ultimate maturity phase combines individual learning artefacts to cover a broader subject area. As a consequence, this subject area becomes teachable to novices, with assessment playing both a formative and, possibly, summative role.

However, the different levels of interaction that accompany this process need to be considered. Here, progression from the level of individuals to communities and organisations, with personal networks and professional communities ensuring that interaction goes beyond the boundaries of particular work organisations, needs to be found. Additionally, the maturing process needs to be framed by the idea of developing different types of knowledge assets that are vital for the learning, working and development in any kind of network or organisation. These assets relate to content, processes and semantics. Content such as documents, images, videos etc. can

clearly play an important role in e-learning. Process development can include reflection and formative assessment in ways that enable recording and sharing of individual work practices. Finally, for the linkage of assets it is necessary to take the semantics into account (how the different assets can support individual and collective learning processes by providing the basis for mutual understanding). This is especially important as we will be facilitating bottom-up development of ideas about effective practice, with practitioners contributing their individual views, experiences and insights. Without semantic integration, such an approach could embed misinterpretations. Overall, this approach to developing a workplace PLE for careers guidance practitioners offers the prospect of deepening and contextualising knowledge and understanding of how to apply a range of technical-communicative skills in careers guidance delivery with a stronger focus upon LMI. Such an approach has processes of critical reflection and formative assessment embedded within it.

A PLE should also support social learning through groups [5] and the mapping of contextual knowledge by the P.A.s. The project aim is not to develop an information environment, but to develop a tool, or set of tools, to allow personal access to resources from multiple sources, and to support knowledge creation and communication. Based on an initial scoping of knowledge development needs, an initial list of possible functions for a PLE have been suggested, including: access/search for information and knowledge; aggregate and scaffold by combining information and knowledge; manipulate, rearrange and repurpose knowledge artefacts; analyse information to develop knowledge; reflect, question, challenge, seek clarification, form and defend opinions; present ideas, learning and knowledge in different ways and for different purposes; represent the underpinning knowledge structures of different artefacts and support the dynamic re-rendering of such structures; share by supporting individuals in their learning and knowledge; networking by creating a collaborative learning environment. It should be noted that these knowledge development activities are not linear and that: one will precede another; people will undertake different groups of activities; activities are discrete; or that one feeds into another. This initial scoping was progressed by developing a scenario of how a PLE tool, or Mashup, might be used within Connexions Kent.

5. The Connexions Kent scenario

The P.A. at the centre of this fictional scenario is recently qualified and new to Kent, working in a school near Sittingbourne. The young person featuring in this scenario is one of many young people who have been referred to the P.A. for a one-to-one interview by the school's careers coordinator. This particular young person is 15 years old (in Year 11 in her school) and does not wish to stay at school to undertake any higher level qualifications, beyond the compulsory school leaving certificate (General Certificate in Secondary Education), usually taken at age 16. She tells the P.A. that she wants to go into the construction industry, to train to become a plumber as she wants to do a practical job. Her father has told her that plumbers are paid well.

Access/search – To respond to the specific query about plumbing as a potential job prospect for this young person, the P.A. needs to identify relevant sources of LMI.

Given that the P.A. is recently qualified and new to the area, this might include, for example: a job description of a plumber; information about the requirements (both qualifications and/or experience) for entry to plumbing; details on relevant regional training courses; data about the workforce share of women in the industry; regional labour demand forecasts for plumbers over the next 2/3 years; and rates of pay for qualified plumbers. During this process (which might involve disparate sources from the internet, other media or professional colleagues), the P.A. will (ideally) apply criteria to enable them to select sources of reliable, quality LMI.

Aggregate & scaffold – The P.A. will need to aggregate the results of their various searches in a meaningful way. For example, there may be two or three sources of information which relate specifically to entry requirements for plumbing, perhaps in different parts of the country or with different companies. Information from some sources may even contradict that from others and any differences will need to be resolved or presented in a way that helps the young person to understand their options (and issues related to risk and uncertainty). Then, there will be a learning process during which new knowledge about plumbing will be integrated with their existing knowledge. This may require the accommodation of new perspectives.

Manipulate – Having identified valid and reliable sources of LMI and started to make sense of (that is, interpreting) the data, a further phase of organising, or rearranging, data may follow. The P.A. may, for example, wish to develop and expand this new knowledge for use in the future with other young people, at different stages of transition, with different needs. For example, they may wish to integrate the new knowledge about construction with other knowledge (previously acquired) about similar occupational roles in other sectors. This knowledge could also be used to support learning about the new vocational Diploma lines being offered in the County from September, 2008 (the introduction of these Diplomas represents a major overhaul of 14-19 education and training and young people, so parents may know even less than normal about possible options available to their children).

Analyse – The P.A. may subsequently wish to expand their newly acquired knowledge about plumbing with other sources of LMI about the construction industry in the local area for different purposes. So, they may wish, for example, to integrate their new knowledge into a group work presentation to Diploma students in Construction. For this, a process of analysis is needed during which the P.A. discusses the requirements for such a session with the relevant tutor on the Diploma course, during which he consolidates his own learning about the construction industry with that of the Diploma tutor. This process of peer discussion may well include discussion with other P.A.s involved in similar activities across the county of Kent.

Reflect – As part of their professional role, P.A.s will routinely reflect on their learning so that they can improve their own performance, with the overall aim of improving the services offered to their clients. These processes of reflection are likely to relate to an established (or even newly stimulated) interest in employment opportunities in construction in the local area in which they operate. This will typically involve questioning and seeking clarification around some aspects of plumbing in particular, and construction in general. This may subsequently result in further research and consolidation of new knowledge into, for example, a written resource to make available to schools and/or students in the locality.

Present – In this scenario, the target for the presentation of the new knowledge about construction will be the young person who is the subject of the original referral. The P.A. will need to make professional judgements about the form and manner in which the newly acquired intelligence about the prospects for local plumbing apprenticeships will be conveyed, together with pay rates and employment prospects.

Represent – A more sophisticated mediation of the newly acquired knowledge may be produced, subsequently, for different purposes: for example, in the development of an information sheet for parents showing different possible learning and development pathways for young people leaving school at 16.

Share – This phase of knowledge maturing depends on the context in which the P.A. is operating. For example, they may have the opportunity to share their emerging understanding, with other P.A.s, labour market specialists or senior managers within Connexions Kent and teachers across various organisations.

Network with other people – Finally, networking will be a key challenge. In contexts where there is an increased awareness of the importance of LMI for the guidance process, together with support from the management to engage in a systematic dissemination of new knowledge, this should occur on a systematic basis. Where this is not occurring formally, it may well be occurring in ways that are more serendipitous and *ad hoc* (for example, over lunch or a coffee break).

6. A Mashup PLE

It should be stressed that the scenario outlined above is only illustrative of how a PLE would be used in practice. As was stated earlier, P.A.s have wider responsibilities than utilising LMI to advise young people about careers and it is intended that the PLE will assist them in the other aspects of their work too. Nevertheless, we believe that the idea of a Mash-up PLE may offer an initial opportunity to support the scenario. The Mashup may take different forms. It may involve bringing together distributed data from different sources and allowing the aggregation and manipulation of that data. It may also involve the use of public web APIs (for instance, for social networking and for providing access to individual resources), and also the creation of tools for authoring, creating and editing artefacts. It will certainly involve the use of services for developing and exposing semantic knowledge structures and supporting P.A.s in developing models and maps of contextual knowledge. It is not argued that a single application is capable or sufficiently flexible to achieve such aims, but rather we view a Mashup PLE as providing the glue to support the different knowledge maturing processes. A further approach could be the development of a PLE Mashup kit to assist the P.A.s in developing their own Mashup PLEs. Furthermore we are aware that many of the knowledge processes described are collaborative and take place within an organisational community. Whilst we are still unclear about the relationship between the PLE and such a wider Organisational Learning and Maturing Environment it would appear that we need a seamless movement between the individual and collective environments, which may be a further facet of services provision or a Mashup approach. The next steps in our work will be to finalise the ethnographic study into the work of the Connexions Kent P.A.s and to refine and

extend the scenario outlined above. This in turn will feed the work in producing an initial mock-up of the PLE leading to a cycle of user centred development work.

References

1. Agostini, A., Albolino, S., Michelis, G. D., Paoli, F. D., & Dondi, R. (2003). Stimulating knowledge discovery and sharing. Paper presented at the 2003 International ACM SIGGROUP Conference on Supporting group work, Sanibel Island, Florida, USA.
2. Attwell, G. (2007). The personal learning environments – The future of eLearning? *eLearning Papers*, 2(1). <http://www.elearningeuropa.info/files/media/media11561.pdf>. (accessed 11/8/08).
3. Attwell, G. & Brown, A. (2000). Knowledge development at the interface of research, policy and practice – support for knowledge development within the CEDEFOP Research Arena (CEDRA), Paper presented at IVETA 2000 conference, Hong Kong, August 6-9th, 2000
4. Brown, A. (1997). A dynamic model of occupational identity formation, in A. Brown (Ed) *Promoting Vocational Education and training: European perspectives*, Tampere, Tampereen yliopisto, (pp 59-67).
5. Brown, J. S., & R. P. Adler. (2008). Minds on fire: Open education, the long tail, and Learning 2.0. *Educause Review* 43 (1): 16-32. <http://connect.educause.edu/Library/EDUCAUSE+Review/MindsonFireOpenEducation/45823> (accessed 13/08/08).
6. Fischer, M. D. (1995). Using computers in ethnographic fieldwork. In R. M. Lee (Ed.), *Information Technology for the Social Scientist* (pp. 110-128). London: UCL Press.
7. Hung, D. (2002). Forging links between ‘Communities of Practice’ and Schools. *Journal of E-Learning*, 1(2), 22-33.
8. Lindstaedt, S., & Mayer, H. (2006). A storyboard of the APOSDLE vision. Paper presented at the 1st European Conference on Technology-Enhanced Learning, Crete (1-4 October 2006).
9. Maier, R., & Schmidt, A. (2007). Characterizing knowledge maturing: A conceptual process model for integrating e-learning and knowledge management. Paper presented at the 4th Professional Knowledge Management Conference, Potsdam, Germany.
10. Nonaka, I., & Takeuchi, H. (1995). *The knowledge creating company. How Japanese companies create the dynamics of innovation*. Oxford: Oxford University Press.
11. Razavi, M. N., & Iverson, L. (2006). A grounded theory of information sharing behavior in a personal learning space. Paper presented at the 20th Anniversary Conference on Computer Supported Cooperative work, Banff, Alberta, Canada.
12. Schmidt, A. (2005). Knowledge Maturing and the Continuity of Context as a Unifying Concept for Knowledge Management and E-Learning. Paper presented at the I-KNOW 2005, Graz.

13. Wilson, S., Liber, O., Johnson, M., Beauvoir, P., Sharples, P., & Milligan, C. (2006). Personal learning environments challenging the dominant design of educational systems. Paper presented at the ECTEL Workshops 2006, Heraklion, Crete (1-4 October 2006).