Mash-Up Personal Learning Environments (MUPPLE’09)
Fridolin Wild, Marco Kalz, Matthias Palmér, Daniel Müller (Eds.)

Workshop in conjunction with the 4th European COference on Technology-Enhanced Learning (ECTEL’09): Synergy of Disciplines
Please refer to these proceedings as


Copyright © 2009 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.
Organisers

Fridolin Wild (The Open University, UK)
Marco Kalz (Open University, The Netherlands)
Matthias Palmér (KTH Royal Institute of Technology, Sweden)
Daniel Müller (IMC AG, Germany)

Programme Committee

Abelardo Pardo (University Carlos III de Madrid, Spain)
Dai Griffith (University of Bolton, UK)
Denis Gillet (EPFL, Switzerland)
Effie Law (University of Leicester, United Kingdom)
Felix Mödritscher (Vienna University of Economics and Business, Austria)
Graham Atwell (Pontydyssgu, 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)
Stephane Sire (EPFL, Switzerland)
Tony Hirst (Open University, UK)
Bernd Simon (Knowledge Markets, Austria)
Jad Najjar (Vienna University of Economics and Business, Austria)
Kai Hoever (IMC AG, Germany)
Zuzana Bizonova (INT Paris, France)
Jan M. Pawlowski (University of Jyväskylä, Finland)
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. These PLEs typically complement or interface with Learning Management Systems (LMS) with additional widgets, services, and data integrated from and with organization-external learning tools.

Consequently, the aim of this workshop is not to discuss the concepts ‘PLE vs. LMS’, but to focus more generally on how learning experiences can be enriched using mash-ups of widgets and services with microformats and how technology can help to respond automatically to competence level, need, or context. Moreover, the investigation of necessary competencies to deploy mash-up technologies is dedicated special attention in this workshop.

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. Interoperability is the enabler to allow these different components to be worked together facilitating the achievement of the underlying learning task.

In a wide range of European IST-funded research projects such as LTFL, Palette, STELLAR, TENcompetence, Mature, iCoper, Role, and OpenScout 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 individuum 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.

Acknowledgements

This workshop is supported by the European Commission as a concertation activity between especially the European Projects LTLL, STELLAR, Role, iCoper, Mature, OpenScout, Palette, and TENcompetence. We especially would like to thank the members of our programme committee listed above.

Fridolin Wild,
Marco Kalz,
Matthias Palmér,
Daniel Müller

Table of Contents

Editorial
Fridolin Wild, Marco Kalz, Matthias Palmér, Daniel Müller ............................... 3

SMashup Personal Learning Environments
Mohamed Amine Chatti, Matthias Jarke, Zhaohui Wang and Marcus Specht ........ 6

Personal Learning Environments for Higher Education: A Mashup Based Widget Concept
Behnam Taraghi, Martin Ebner and Sandra Schaffert ................................. 15

ReMashed - Recommendation Approaches for Mash- Up Personal Learning Environments in Formal and Informal Learning Settings
Hendrik Drachsler, Dries Peceu, Tanja Arts, Peter van Rosmalen, Hans Hummel, Rob Koper, Edwin Hutten and Lloyd Rutledge ................................. 23

Mapping Web Personal Learning Environments
Matthias Palmér, Stéphane Sire, Evgeny Bogdanov, Denis Gillet and Fridolin Wild .. 31

Moodle Wave: Reinventing the VLE using Widget technologies
Scott Wilson, Paul Sharples, Dai Griffiths and Kris Popat ............................. 47
Towards Collaborative Portable Web Spaces
Stéphane Sire, Matthias Palmér, Evgeny Bogdanov and Denis Gillet .......................... 59

Infusing Reflective Practice in eLearning Courses - Can Widgets Help?
Dominique Verpoorten, Wim Westera and Marcus Specht ................................. 72

The MUPPLE Competence Continuum
Joanna Wild, Fridolin Wild, Marco Kalz, Marcus Specht and Margit Hofer .............. 80

Reflection Support Using Multi-encoded Tag-clouds
Christian Glahn, Marcus Specht and Rob Koper ............................................. 89

Conceptual Design of EduFeedr - an Educationally Enhanced Mash-up Tool for Agora Courses
Hans Poldoja and Mart Laanpere ........................................................................ 98

RDFa versus Microformats: Exploring the Potential for Semantic Interoperability of Mash-up Personal Learning Environments
Vladimir Tomberg and Mart Laanpere ............................................................. 102

Interoperability Issues for Formal Authoring Processes, Community Efforts, and the Creation of Mashup PLE
Roland Klemke and Birgit Schmitz ................................................................. 110

Generalized Model for Interoperability of Data Based on Model Driven Architecture
Zuzana Bizanova ............................................................................................... 114

Differentiating and Defining Portlets and Widgets: A Survey Approach
Effie Law, Daniel Müller and Anh Vu Nguyen-Ngoc ........................................... 123
SMashup Personal Learning Environments

Mohamed Amine Chatti†, Matthias Jarke†, Zhaohui Wang†, and Marcus Specht‡

†Informatik 5 (Information Systems), RWTH Aachen University
{chattijarke,wang}@dbis.rwth-aachen.de,
‡Open University of the Netherlands, Netherlands
marcus.specht@ou.nl

Abstract. Mashups have become the driving force behind the development of Personal Learning Environments (PLE). Creating mashups in an ad hoc manner is, however, for end users with little or no programming background not an easy task. In this paper, we leverage the possibility to use Semantic Mashups (SMashups) for a scalable approach to creating mashups. We present the conceptual and technical details of PLEF-Ext as a flexible framework for mashup-driven end-user development of PLEs. PLEF-Ext uses the Service Mapping Description (SMD) approach to adding semantic annotations to RESTful Web services, and leverages the SMD annotations to facilitate the automatic data mediation and user-friendly creation of learning mashups.

1 Introduction

Recently, the mashup concept has emerged as a core technology of the Web 2.0 and social software movement. A mashup indicates a way to create new (Web) applications by combining existing data and services from several sources. In the past few years, Web mashups have become a popular approach towards creating a new generation of customizable Web applications. The popularity of Web mashups has mainly been driven by the increasing popularity of lightweight RESTful Web services, AJAX, and JSON that build core technologies in the Web 2.0 movement. Mashup development has also become a very popular re- search topic in TEL. Many TEL researchers recognized the value of mashups and have already adopted the mashup concept for Personal Learning Environment (PLE) development. The idea behind mashup PLEs is to let learners create their very own learning mashups that leverage components and content generated by learning service providers and other learners around the Web. Yet, most of the proposed mashup PLE approaches are based on widgets. In this paper, we go a step further toward supporting the mashup of RESTful services and the use of semantic approaches to deal with service integration and mediation within mashup PLEs. We propose a mashup PLE framework, called PLEF-Ext, that enables a user-friendly creation, management, sharing, and reuse of learning mashups.

The paper proceeds as follows. In Section 2, we discuss how mashups can provide a powerful tool to build PLEs. In Section 3, we provide an overview of some popular and representative mashup development tools and frameworks, and summarize the main problems in mashup development today, focusing specifically on the data mediation and integration challenges. In Section 4, we highlight the benefits of using semantic mashups for a scalable and flexible mashup development. Section 5 presents the Semantic Mapping Description (SMD) approach to adding semantic annotations to RESTful Web services and shows the benefits of adopting a semantic mashup approach based on SMD. We follow in Section 6 with the conceptual and technical details of PLEF-Ext, a mashup PLE framework that can help learners share, find, integrate, reuse, and easily remix learning services based on a semantic description of the same. And finally, we summarize our findings in Section 7.

2 Mashup Personal Learning Environments

A Personal Learning Environment (PLE) is a learner's gate to knowledge. From a technical point of view, a PLE can be viewed as a self-defined collection of services, tools, and devices that help learners build their Personal Knowledge Networks (PKN), encompassing tacit knowledge nodes (i.e. people) and explicit knowledge nodes (i.e. information). Thus, mechanisms that support learners in building their PLEs become crucial. Mashups provide an interesting solution to developing PLEs. We differentiate between two types of mashups:

6
- **Mashups by aggregation** simply assemble sets of information from different sources side by side within a single interface. Mashups by aggregation do not require advanced programming skills and are often a matter of cutting and pasting from one site to another. Personalized start pages, which are individualized assemblages of feeds and widgets, fall into this category.

- **Mashups by integration** create more complex applications that integrate different application programming interfaces (APIs) in order to combine data from different sources. Unlike mashups by aggregation, the development of mashups by integration needs considerable programming expertise.

Most of the state-of-the-art mashup PLE solutions only focus on the first type of mashups, i.e. mashups by aggregation. These solutions mainly support learners in juxtaposing content from different sources (mainly feeds and widgets) into a single interface. Examples include PLEF\(^1\), MUPPLE\(^2\), and - although not designed as educational technology - Personalized Start Pages such as iGoogle\(^3\) and Netvibes\(^4\). In the next section, we discuss the current status of mashup development with a focus on the second type of mashups, i.e. mashups by integration.

### 3 Mashup Development

The Internet and its related technologies have created an interconnected world in which we can easily exchange and reuse information in unforeseen, unexpected ways. Web services are emerging as a major technology for deploying automated interactions between distributed and heterogeneous applications [2]. In the Web 2.0, services based on the representational state transfer (REST) paradigm [3] are increasingly popular for Web development. RESTful services often take the form of RSS/Atom feeds and AJAX based lightweight services, which explains their greater success compared to heavyweight services, which are based on the Web Services Description Language (WSDL) and SOAP. The output formats of RESTful services, often in the form of Extensible Markup Language (XML) or the more lightweight form JavaScript Object Notation (JSON), make RESTful services ideal for AJAX-based mashups.

Developing mashups is however not an easy task. In general, mashup development is still very much an ad hoc activity. Mashups are often generated manually using normal Web development technologies such as HTML, CSS and JavaScript [4]. Creating a mashup in a manual manner is a very time consuming task and impossible for the typical Web user [5]. A key difficulty in creating mashups is data mediation between the services to be mashed up. Typically, in order to create a mashup, the user would need to understand not only how to write code but also the APIs and descriptions of data formats of all the services that need to be included in the mashup [6].

To lower the barrier of creating a Web mashup, leading companies are now actively developing different mashup building tools and platforms, that require little to no programming knowledge from the user. Yahoo! Pipes\(^5\), Microsofts Popfly\(^6\) and Google Mashup Editor\(^7\) are some well-known examples of mashup platforms that users have largely adopted. In general, these platforms provide a higher level of abstraction and use visual programming techniques to facilitate the creation of mashups. For instance, Yahoo! Pipes provides a visual editor to wire different graphical elements to each other, based on different kinds of data processing operations such as regular expressions,

\(^{1}\) [http://eiche.informatik.rwth-aachen.de:3333/PLEF/index.jsp](http://eiche.informatik.rwth-aachen.de:3333/PLEF/index.jsp)

\(^{2}\) [http://mupple.org](http://mupple.org)

\(^{3}\) [http://www.google.com/ig](http://www.google.com/ig)


\(^{5}\) [http://pipes.yahoo.com/](http://pipes.yahoo.com/)

\(^{6}\) [http://www.popfly.com/](http://www.popfly.com/)

\(^{7}\) [http://code.google.com/gme/](http://code.google.com/gme/)
filters, sorting or looping instructions. Similarly, Microsoft Popfly provides a Web-based GUI, where
the user can connect different Web services to each other by dragging the blocks representing them
into the main display and drawing wires to connect those services. Google Mashup Editor (GME)
provides a higher level JavaScript API for manipulating data programmatically. A detailed description
and comparison of these and other mashup development tools is provided in [7].

Google Mashup Editor has not been designed to be used by end users without background in
programming. It is more suitable for amateur and professional mashup programmers. And, other
mashup editors, such as Yahoo! Pipes and Microsoft Popfly, that are intended for non-professional
mashup developers suffer from two main limitations.

First, most of the mashup tools are restricted to services that have standard types of outputs, such as
RSS or ATOM. These mashup tools are particularly capable at processing and creating new feeds. A
common feature is (1) pull data from external RSS and Atom feeds, (2) filter, sort, and combine many
feeds into one, (3) grab the output as RSS, JSON, KML, and other formats, and (4) include the result as
a widget in other Web sites.

Second, the usage scope of these mashups tools is often limited to services that are internal to the
company where the mashup tool was developed (Yahoo! Pipes, for example, adds Yahoo! Maps to any
pipe containing GeoData). And, in most cases these mashup tools are restricted to a set of few of
popular services such as Flickr, Digg, or popular map, traffic or weather services.

As a consequence, a plethora of existing (RESTful) services cannot be used by these mashup tools. If
one of the companies behind the aforementioned mashup tools, for some reason, would be willing to
add a new external service to their supported service pool, it would be necessary to implement a new
wrapper in order to accommodate the new service. This is, however, not a scalable solution due to the
rate at which new services are coming online [6]. Moreover, it would be extremely difficult to
work with a service whose inputs and outputs are in a different format. This makes it difficult to
address issues related to data interoperability, integration, and mediation [5].

The concept of Semantic Mashups (SMashups) [8] addresses these limitations by proposing the
semantic annotation of Web services as a solution to the service integration and mediation
challenges.

4 Semantic Mashups (SMashups)

The challenges of service interoperability, reuse, integration, and mediation have led to several
proposals for Semantic Web services. Several research projects have looked at semantics for
traditional (WSDL or SOAP) Web services to help address heterogeneity, reuse, and mediation
challenges, and the community took a step toward supporting semantics for Web services by
adopting Semantic Annotation for WSDL (SAWSDL)\(^8\) as a W3C recommendation in 2007. The
SAWSDL specification enables semantic annotations for Web services using and building on the
existing extensibility framework of WSDL [2, 9]. In SAWSDL, se-
\(^8\) http://www.w3.org/2002/ws/sawSDL/

mantic annotations describing inputs, outputs, operation, interfaces, and faults, are embedded as properties in the WSDL [5].

Recently, driven primarily by the popularity of lightweight RESTful Web services, AJAX, and JSON
that build core technologies in the Web 2.0 movement, attention has shifted to using semantics to
annotate RESTful Services. For instance, [5, 6] proposed a framework called Semantic Annotation of
REST (SA-REST) to add semantics to RESTful services. SA-REST builds upon the authors’ original
ideas in WSDL-S [10], which was the primary input of the W3C recommendation SAWSDL.

The key difference between SAWSDL and SA-REST is that unlike SAWSDL, where annotations are
added directly in the WSDL of the service, SA-REST annotations have to be embedded into the HTML page that describes the service. Consequently, SA-REST uses microformat-based approaches such as RDFa\(^9\) and Gleaning Resource Descriptions from Dialects of Languages (GRDDL)\(^10\) to add and capture annotations via properties of HTML elements [6].

The major drawbacks of an SA-REST approach to add semantic annotations to RESTful services are threefold:

First, semantic annotation with SA-REST requires the use of microformats. Microformats, however, come in many different competing forms. And, even the process of adding semantics with standardized microformats such as RDFa or GRDDL is not straightforward. For instance, to annotate a HTML page with GRDDL the service provider needs (1) to embed the annotations in any, not necessarily standardized, microformat, (2) add the URL of the GRDDL profile to the head element in the HTML document, and (3) add to the head element a link tag that contains the URL of an XSLT that translates the GRDDL profile into SA-REST specific RDF triples [5]. The complexity of these microformats makes it hard for SA-REST to be adopted by service providers for the semantic annotation of their RESTful services.

Second, the annotation of a concept (e.g. input, output, operation) in SA-REST is a way to tie together the concept to a class that exists in an ontology [5]; SA-REST, thus, assumes an agreement among all service providers on a common ontology. This is however not realistic, since it would be impossible to get all service providers to agree on a single ontology to describe their concepts.

Third, besides linking a concept to an ontology class, data mediation in SA-REST requires the specification of a lifting and lowering schemas, that is mapping of the data structure that represents the input and output of a given service to the data structure of the ontology. The use of lifting and lowering schemas means, on the one hand, extra complicated work for the service provider to work with XSLTs or XQueries to specify and keep up-to-date its lifting and lowering schemas, and, on the other hand, higher server workload and additional server-side code in order to deal with parsing and applying the lifting and lowering mappings between the services to be mashed up and the ontology.

Recently, Kris Zyp [11] has proposed a more lightweight and standards-based approach to adding semantics to RESTful services, called Semantic Mapping Description (SMD).

5 Service Mapping Description (SMD)

In this section, we discuss the Service Mapping Description (SMD) approach to adding semantic annotations to RESTful Web services. SMD is a flexible and simple JSON representation describing Web services. A wide array of web services can be described with SMD including RESTful services and JSON-RPC services. SMD uses JSON Schema to provide a definition for a variety of available services, and document how to call the services, what parameters are expected, and what to expect in return [11]. Figure 1 shows the Yahoo! search service semantically annotated with SMD.

Similar to SA-REST, SMD facilitates data mediation between the services to be mashed up, since it provides a thorough description of how a service can be invoked, the type of data that should be passed to the service and what will be returned from it. The main advantage of SMD, as compared to SA-REST, is that it is based on JSON. JSON is a lightweight data interchange format whose simplicity has resulted in widespread use among Web developers. JSON offers two major advantages over XML. First, JSON-encoded data is less verbose than the equivalent data in XML. XML uses duplicate start and end tags to wrap data values. A JSON object, by contrast, is simply a series of comma-separated name:value pairs wrapped in curly braces. This yields better performance of the JSON-based Web

\(^9\) http://www.w3.org/TR/xhtml-rdfa-primer/

\(^10\) http://www.w3.org/TR/grddl/
applications because JSON data downloads more quickly than XML data.

JSON’s second benefit is that it’s easy to parse using any programming language, and its structures map more directly onto the data structures used in modern programming languages. Because it is essentially a string representation of a JavaScript object (hence the name), browsers can parse JSON data simply by calling the JavaScript "eval" function. Any field of the JSON object can then be directly accessed by name.

```json
"SMOVersion": "2.0", // version level of the SMD being used
"transport": "JSONP", // transport mechanism; can also be "POST", "GET", "REST", or "TCP/P"
"envelope": "URL", // how a service message string is created from the provided parameters

"target": "http://yahoo.com",
"additionalParameters": true,
"parameters": [{
"name": "output", "optional": false, "default": "json"}, ...
],
"callbackParamName": "callback",
"services": {
"Yahoo Web Search": {
"target": "http://search.yahooapis.com/WebSearchService/V1/webSearch", // URL for method call requests
"parameters": [{
"name": "query", "type": "string", "optional": false,
...
],
"return": {
"type": "object",
"properties": {
"ResultSet": {
"type": "object",
"properties": {
"Result": {
"type": "array",
"items": {
"type": "object",
"properties": {
"Title": {"type": "string"},
"Summary": {"type": "string"},
"Url": {"type": "string"},
...
}"
}"
...

Fig. 1. Semantic Annotation of RESTful Web Services with SMD

JSON is also the basis of a very powerful technique for building client-based (in-browser) mashups, namely JSON with Padding (JSONP), which was created as a workaround to the Same-Origin Policy (SOP) problem. In Web 2.0, Asynchronous JavaScript and XML (AJAX) is becoming the driving force behind mashups. In AJAX, data is retrieved using the XMLHttpRequest function, which is an API that lets client-side JavaScript make HTTP connections to a server. The SOP, however, prevents the AJAX code from making an XMLHttpRequest call to a URL that is not on the same domain as the current page, because of the cross-domain attacks that could result.

The normal way around SOP is to have the Web server behave as a proxy. The Web page then requests data from the Web server it originates from, and the Web server handles making the remote call to the third-party server and returning the results to the Web page. Although widely used, this technique isn’t scalable.

Another way to overcome the SOP limitation is to insert a dynamic script element in the Web page, one whose source is pointing to the service URL in the other domain and gets the data in the script itself. It works because the same-origin policy doesn’t prevent the insertion of dynamic script
elements into the Web page and treats the scripts as if they were loaded from the domain that provided the Web page. This is actually how JSONP works: load the JSON response within a <script> tag. In order to do this, the name of a callback function is specified as an input argument of the call itself. The remote server will then wrap the JSON response in a call to that function. When the browser finishes downloading the new contents of the <script> tag, the callback function executes. This approach requires the remote Web service to behave as a JSONP service; that is a Web service with the additional capability of accepting a callback function name as a request parameter and supporting the wrapping of the returned JSON data in a user-specified function call [1]. Popular service providers such as Google and Yahoo! already support this technique.

The simplicity of JSON as a lightweight data interchange format and the power of JSONP as a flexible technique for client-based (in-browser) mashups, make SMD much better suited to SMashups than SA-REST.

6 PLEF-Ext: SMashup PLEs with SMD

PLEF-Ext provides a flexible framework that can help learners easily extend their PLEs with new learning services. Driven by the SMashup concept, PLEF-Ext supports learners in sharing, finding, integrating, managing, reusing, and remixing semantically annotated RESTful learning services with minimum effort. PLEF-Ext uses SMD for the semantic description of the services. An abstract architecture of PLEF-Ext is provided in Figure 2. PLEF-Ext encompasses five main modules:

- The Service Discovery Module (iSearch) provides a tag-based search mechanism that enables learners to locate learning services remixed and shared by peers within PLEF-Ext.
• The Service Integration Module (iPull) makes it possible for the learner to easily plug in a new SMD-annotated external learning service into her PLE, by just specifying the URLs of the service and its SMD.
• The Service Adaptation Module (iAdapt) enables the learner to adapt a service to her needs, by just specifying the parameters and return values of the service.
• The Service Aggregation Module (iMix) lets learners compose a mashup query in a drag-and-drop fashion.
• The Service Management Module (MyStuff) supports the learner in managing (i.e. add, remove, edit, tag) her learning services within PLEF-Ext.

PLEF-Ext is built upon a client-server architecture. However, in contrast to traditional mashup environments, where the majority of computation-intensive processing will occur on the server (mashups are created using server-side proxies), the client in PLEF-Ext is much fatter. That is, the mashups will be built and run on the client-side, using the computing power of the client device. Coupled with JSONP services that expose their data in JSON and wrap it in function call, the client (browser) can make requests for data from the different services and use the fetched data to build in-browser mashups. In order to achieve this, the client in PLEF-Ext is assisted with an extensive user interface library of widgets and panels provided by SmartGWT\textsuperscript{11} and powerful GWT\textsuperscript{12} libraries to perform a wide range of actions, such as parsing of the fetched JSON data and asynchronous server communication through HTTP or remote procedure calls (RPCs).

Fig. 3. SMashup Process in iMix

\textsuperscript{11} http://code.google.com/p/smrtgwt/
\textsuperscript{12} http://code.google.com/webtoolkit/
In the following, we discuss in more details the functionalities of the Smashup module iMix. Figure 3 demonstrates the Smashup process in iMix. We suppose that the services to be mashed up are JSONP services. Their SMDs, however, are available on remote servers that return pure JSON data without wrapping it in a function call (since SMD is a relatively new concept, it is not supported yet by third-party service providers. For this work we are using SMDs that we created for popular JSONP services such as Google data APIs, Yahoo! search, geonames, delicious, and flickr). In this case, in order to work around the SOP security problem, we need to create a proxy on our local server. To note that if third-party service providers would in the future adopt SMD and would be able to return the SMD data via JSONP, the proxy server can be omitted and interaction with remote resources will be done from the client-side, solely via JSONP. This would then make the PLEF-Ext architecture much simpler. In order to mashup two RESTful Web services with iMix, a learner first needs to select the services to be mashed up. The client then makes HTTP calls (XMLHttpRequests) to the local server (proxy server) and have it go fetch the SMD data from the remote servers (step 1 in Figure 3). This can be achieved in GWT either with GWT RPC or direct HTTP using GWT RequestBuilder. The local server downloads the JSON-encoded SMDs from the remote servers (step 2) and passes it back to the client (step 3). When the client retrieves the SMDs from the local server, a JSON parser (GWT JSONParser) is applied to extract the descriptions of the services from their SMDs (i.e. service inputs, outputs, and request type) and display the descriptions to the learner in the Smashup Editor. The Smashup Editor is the key component of iMix. As mentioned earlier, a key difficulty in creating mashups is data mediation between the services to be mashed up. The main role of the Smashup Editor is to use semantic annotations to enable automatic data mediation without any programming knowledge from the learner side. The learner would only need to perform drag-and-drop actions using the Smashup Editor to define the smashup steps: (1) construct the query URL at which to invoke service 1, (2) construct the query URL at which to invoke service 2 and specify the mashup parameters (e.g. the outputs of service 1 that should be inputs for service 2), and (3) specify the outputs of service 2 (i.e. mashup result). Once the smashup steps have been defined, the learner can start the execution of the real mashup. At this point, the client invokes service 1 via JSONP to gather the data that are needed for the mashup (step 4 and 5) and invokes service 2 via JSONP to get the final result of the mashup (step 6 and 7), which will be shown to the learner in the mashup viewer (step 8).

7 Conclusions

In this paper, we discussed the challenge of PLE extensibility and how mashups can provide a powerful tool to achieve this challenge. We outlined the limitations in current mashup development platforms and tools, and discussed the benefits of Semantic Mashups (SMashups) as a flexible and scalable approach to creating mashups. We also discussed the Service Mapping Description (SMD) approach to adding semantic annotations to RESTful Web services. We finally presented the conceptual and technical details of PLEF-Ext as a mashup PLE framework that leverages SMD-based semantic annotations of RESTful Web services for user-friendly learning mashups. All of these efforts have been made available on the PLEF-Ext project homepage13. We welcome feedback from colleagues and users on both their experiences with the system and new ways they would suggest to leverage semantic mashup techniques to help learners build and extend their PLEs.

References


13 http://eiche.informatik.rwth-aachen.de:3333/PLEF-Ext/
Personal Learning Environments for Higher Education: A Mashup Based Widget Concept

Behnam Taraghi¹, Martin Ebner¹, Sandra Schaffert²

¹Graz University of Technology, Social Learning, Computer and Information Services, Graz, Austria
b.taraghi@tugraz.at
martin.ebner@tugraz.at

²Salzburg Research, Information Society Research, Salzburg, Austria
sandra.schaffert@salzburgresearch.at

Abstract. Due to the enormous growth of distributed applications, services, tools and resources, it is not easy for end users to come across existing services, manage and use them in a matter which is customized according to their personal needs. Mashups can be a very interesting approach to overcome challenges of distributed (unknown) services. Using mashups in a Personal Learning Environment (PLE) can help to connect resources and applications in one environment customized to the needs of individual users. In this paper a first approach and the concept of a PLE especially for higher education is described. The technological concept is introduced and design issues are pointed out as well as the first prototype is described.

Keywords: PLE, Personal Learning Environment, Widget, Mashup, e-learning

1. Introduction

Since Tim O’Reilly [1] pronounced for the first time the so-called Web 2.0 the relation between people and their content changed dramatically. Interactivity strongly increased Sharing and collaborating by using social software has become a common task since then. The ubiquitous availability and pervasive use of the WWW dominates more and more our social life, our working environment as well as our learning and teaching behaviors [2], [3]. Downes defined the use of Web 2.0 technologies as e-learning 2.0 [4] and numerous research works carried out, how it can be used within different didactical settings [5]. Disregarding the type of web based software – Wikis [6], Weblogs [7] or Podcasts [8] – the integration of these establish new potentials for creating new learning environments. Nowadays not only social software such as Facebook or Twitter gain more and more importance [9], but also platforms for sharing different kind of media such as YouTube (video), Slideshare (presentations)
or Del.icio.us (bookmarks) are an integral part of innovative teaching approaches and strengthen informal learning [10].

Due to the enormous and fast grow of different applications for different purposes it can be stated that managing such tools becomes more and more sophisticated. It is easy imaginable that teachers as well as learners were quickly swamped by these possibilities. Different studies on the use of Web 2.0 technologies by university freshman pointed out that a lot of (popular) applications are even not known yet [11].

Particularly to overcome the challenge of distributed and potentially unknown resources as well as tools and the related information overload, mashups become interesting [12]: The possibility to connect different resources from different Web servers within one environment should help to maintain overview of all activities. mashups merge content, services and applications from multiple Web sites in an integrated, coherent way [13]. Concerning learning issues, mashups are seen as precondition for a new form of a personalized learning environment [14].

At Graz University of Technology (TU Graz) first steps have been done towards a Personalized Learning Environment (PLE) for higher education institutions. In this publication the need for PLEs, the technological concept, the first layout as well as the first running Graphical User Interface (GUI) is described. Bearing the multiple possibilities of meaningful integration of services and applications offered by a university in mind, we aim to provide our first research results.

2. The Genesis and Idea of “Personal Learning Environments”

The challenge and possibility to connect and mashup different Web based applications were the cradle for a new concept in field of technology enhanced learning: The idea of a “Personal Learning Environment” comes up [15]: Existing technological concepts of learning and teaching in the Web, such as Learning Management System (LMS), are mainly developed to support managerial teaching needs such as student management and course organization. In contrary, the PLE concept takes the individual learner and his/her personal learning interests serious: Within a PLE (s)he arranges and uses Web (learning) resources and Web based (learning) tools in a way that it supports her/his personal knowledge management and learning.

Not surprisingly, the concept of PLE is a young one [15]: One of the first who described the idea of a personal learning environment had been Olivier and Liber (2001) [16]. Some years later, Wilson (2004) [17] sketched an image of a future “virtual learning environment” integrating also external services and applications. Starting in 2006, PLE starts to get more and more attention, as an analysis of the occurrence of the search term “personal learning environment” in Google Insights illustrates [18] and several publications points out [19], [20].

According to the current definitions of PLE, they can be described as learning applications, where the learners can integrate and organize distributed online information, resources and contacts as well as also to provide content and other things developed in the PLE for other (external) online environments [15].
PLEs are not the first approach to personalize learning content. In contrary, there is a long tradition of Instructional Design and adaptive learning, building on so called intelligent systems. Instructional Design follows the idea of the possibility to foster learning, in well dosed, sequenced instruction bits. Particularly in the domain of artificial intelligence the possibility of automatic “personalization” of the content is considered as an automatic adaptation of the learning content by the system, pre-defined by an expert model. For several reasons, these former ideas seem to be out of date or not fitting [21]: (i) learning content is mainly dynamic, permanently under development and only shallowly categorized (miscellaneous); (ii) referring to current learning theory the learner is to be seen as an active, self-organized creator of his/her learning environment and (iii) social involvement and interaction is crucial for learning. Additional arguments for the new forms of personalized learning focus on new ideals and ideas of learning as stated in [14]: learning to learn is more important than (re-) constructing domain-specific knowledge, therefore the establishment of a (networked) learning environment can already be seen as learning outcome. From a more pragmatic point of view, a system that was built on emergence should be more powerful than “programming” by rules (ibid.).

Due to new possibilities, perspectives, insights and challenges the concept of PLE seems to be an interesting, but not very well developed or elaborated concept for an innovation within technology enhanced learning; also in higher education.

3. Technological Concept

The students should be empowered to manage and organize their learning environments personally according to their own needs. This could reduce the management overhead and resource negotiations while enhancing the self-management, self-organization and self-coordination of students [31]. The customization of GUI is important to give learners a personalized look-and-feel. Besides that, the users should be able to use and customize the various decentralized learning applications and university services on the Web in their PLE. In order to meet these requirements a mashup of widgets can be used. Widgets are small embedded applications that can be included in HTML pages. The detailed description of widgets can be found in [23]. A PLE realized as a Rich Internet Application (RIA) and widget container obtains a flexible design structure which would simplify the design customization enormously. On the other hand a mashup of different widgets can serve as front-end applications for distributed learning objects and services in a mashup based PLE. Hoyer [24] introduces some existing mashup tools with different emphasis such as Yahoo Pipes and Microsoft Popfly. Aumüller and Thor [25] describe three main components of a mashup application: Data extraction, data flow and presentation. They categorize different mashup tools to one or some of these components. At TU Graz a PLE is being developed that resembles the vision of “Future VLE” described in [32]. It bases on a mashup of widgets and can be classified to the end-user mashups as described in [26]. It contains a widget engine, implemented in IST Palette project [27] to load and handle the widgets according to the W3C Widget 1.0 Specifications. While the data extraction is carried out on the
server side, the data flow and presentation components are handled by the widget engine on the client side as depicted in figure 1.

The W3C 1.0 family of specifications consists of some specifications which is a step forward to a standard for widgets. The W3C Widgets 1.0 Packaging and Configuration [23], for instance, introduces a zip-packaging format, containing the widget source files. It specifies the folder structure, the XML-based manifest file and the mandatory as well as none mandatory elements. The W3C Widgets 1.0 APIs and the Events Specification [28] deal with the functionality of widgets. It defines some APIs to access the meta data in widget configuration file and events related to the view state of widgets.

The Palette Service portal which is implemented by IST Palette project [27] acts as a widget container and contains a widget engine to install, load and handle the widgets according to [23] and [28]. The widget manifest file described in [23] is extended to some additional preferences to facilitate widget customization by the users. This feature is required in a PLE to achieve personalization. There are also some new extensions related to [28] to realize interactions between widgets such as a drag and drop effect. This enables the data flow between widgets which is needed in a mashup tool. The ability to also load remote widgets in Palette platform makes it possible to use distributed learning services which are generated dynamically on remote servers. According to these features the widget engine of Palette is used as the basis for the designed PLE at TU Graz.

4. Design Issues and Technologies

As described above, the Palette project [27] provides a widget engine that is used in PLE development. There are some performance issues that are overseen in Palette project. Special attention is paid to these issues during this work.

All CSS and JavaScript files have to be packed or minimized dynamically to reduce the load size. Using GZIP compression could speed up the JavaScript code if the browser accepts such encoding. All minimized CSS and JavaScript files are put dynamically into one output buffer respectively to reduce the number of browser requests. It means the browser only needs one request for each CSS and JavaScript content to load. On the other hand the CSS and JavaScript libraries which are not required at the start of the application are loaded later on demand per Ajax-request to decrease the load time at the beginning. Caching is another important issue which shall not be overseen.

For the sake of cross-browser compatibility the last jQuery library\(^1\) and jQuery UI versions are used. They are lightweight, CSS 3 compliant and support most common used browser versions. The jQuery library provides based on a unique CSS framework a ThemeRoller tool and a theme gallery which makes customizing the look and feel of the PLE very fast and easy.

Another important issue is the possible use of JavaFX\(^2\). The widgets body can be implemented by JavaFX instead of pure HTML and JavaScript. The great advantage

---


of this technology is the Drag-to-Install feature of JavaFX which enables the user to
drag the widget out of PLE and drop it onto the desktop as a standalone desktop
application. This could be a step towards the realization of the widgets according to
the widget definition in [23].

![Mashup structure of PLE describing data extraction from distributed
resources, data flow between widgets and presentation components in PLE.](image)

**Fig. 1.** Mashup structure of PLE describing data extraction from distributed
resources, data flow between widgets and presentation components in PLE.

### 5. First Prototype

All design aspects and the main GUI elements of the PLE are described in detail in
our paper at ICL09 [29] for further reading. The implementation of the first prototype
is still in progress. The goal is to offer a logged-in user a mashup of widgets
containing all kind of university services and information personally required by that
user. The many services and information distributed on different university portals
and institutes are centralized and offered in one mashup portal. The scenario could be
as follows: The student opens a widget representing the list of courses he is already
registered in. Selecting a course would invoke another widget containing the e-
learning materials and the eventual Podcasts related to that course. If wanted, some
supplementary widgets, such as a time table and a location widget can be auto-
synchronized in the background to offer information on the place and time schedules
of the course. An alert widget can inform the student as soon as a course is canceled
or a new notification is received. The role of widgets is not restricted to the university
services. They can include any kind of learning objects or remote applications
existing in the WWW. The content can be implemented using any conventional
methods such as HTML and JavaScript, Flash, Java applets or JavaFX.

### 6. Discussion

During the design study, the usability tests [29] as well as the development of the first
prototype of the PLE different crucial factors appeared:
(i) Personalization: The first crucial factor deals with the question „Can a system basing on mashups support the idea of a Personal Learning Environment?“ The answer is divergent. On the one hand personalization means full control by the end-user or with other words (s)he has to choose the tools needed for learning. On the other hand from the teacher’s perspective learning occurs within a specific didactical setting and with the aid of appropriate tools. Due to these facts the described prototype should provide a huge amount of possibilities but not too many.

(ii) Individuality of the interface: The usability study [29] pointed out that a complete changeable interface would not be an optimal solution for especially novice users. At least at a first level learners must become familiar with the concept of distributed applications combined by a complex mashup system. An important principle is to reduce the overload at the beginning, to make users comfortable with the environment. As a consequence users will get predefined interfaces which can be adapted incremental appropriate to their needs and knowledge.

(iii) Learning environment and Information environment: A serious discussion must be started whether a PLE supports learning processes or helps to refine information on the Web more according to specific needs or preferences. It will be necessary to provide students as well as teachers general information about their university or their booked courses. In this case the environment acts as clearly arranged information environment. But furthermore the environment serves also as a learning one if it used for learning purposes including any different resources. The mashup principle allows acting as both.

(iv) Flexibility destroys design: The last aspect points out the controversy that during the paper based mock-up process and the first sketches the flexibility of using widgets in any size and at any place is critical for the Web design of the application. The design strongly follows concepts of usability to enhance the handling as well as the appearance of the environment. If end-user can do anything, especially resizing applications dimensions, this principle cannot be fulfilled. As a consequence widgets can only be arranged within a predefined grid.

7. Open Questions and Next Steps

It can be concluded that Personal Learning Environments are a big and important step towards the future of technology enhanced learning, but a lot of future research must be carried out.

Concerning our own concept and the idea of PLE in general we need to understand how learners deal with it and how teachers can implement it within their didactical approaches. As PLE should give appropriate support for learners, including recommendation for different potential learning resources as content, tools, people (e.g. other learners or experts). This is a quite new challenge for research and implementation (e.g. [30]). We have to work out the way of how we can support this in our PLE solution. From the technology point of view it is recommended for the future to apply HTML 5 features in a mashup implementation. The drag & drop functionality between widgets and event handling for data flow can be realized in HTML 5. Another issue is the HTML 5 offline cache possibility which provides tools
to implement the same features as Google Gears has provided to store data on the client side. If this applied, some parts of the mashup portal can run offline too.

In this publication we discussed the concept of a Personal Learning Environment from a specific point of view: How can a mashup concept work within a higher education institution? The design study as well as the prototype [29] carried out that nearly endless individuality assumes experienced users or in case of learning environments learners. We suggest that learners must be led to this expertise step by step. Furthermore from an educational institution’s perspective we have to support applications that are proven as appropriate for learning processes. However the next step will be to roll out the prototype and merge the World Wide Web with the university.

References

ReMashed – Recommendation Approaches for Mash-Up Personal Learning Environments in Formal and Informal Learning Settings

Hendrik Drachsler¹, Dries Pecceu², Tanja Arts², Edwin Hutten², Lloyd Rutledge², Peter van Rosmalen¹, Hans Hummel¹ and Rob Koper¹

Open University of the Netherlands, ¹Centre for Learning Sciences and Technologies & ²Computer Science Department, PO-Box 2960, 6401 DL Heerlen, The Netherlands
{hendrik.drachsler, lloyd.rutledge, peter.vanrosmalen, hans.hummel, rob.koper}@ou.nl, {pecceu, ekh.hutten, tg.arts}@studie.ou.nl

Abstract. This article presents the ReMashed system that recommends learning content from emerging information of a Mash-Up Personal Learning Environment. ReMashed offers advice to find most suitable learning content for individual competence development of lifelong learners. The ReMashed system was initially designed to offer navigational support to lifelong learners in informal learning settings. In this article we want to discuss its ability to be used also in formal learning settings. For this purpose, we discuss the use of two different recommendation approaches for formal and informal learning within ReMashed.

Keywords: recommender system, mash-up, personalization, personal learning environments, MUPPLE, informal learning, emergence, formal learning

1 Introduction

The networked Knowledge Society is more than ever empowered by the Web2.0 development efforts. The so called Web2.0 lifted the barrier of adding information to the Internet and enables people to contribute information to the Internet. This also applies to lifelong learners who use the Internet to find suitable information for their learning needs. They also take advantage of Web2.0 tools to create, share, and use learning activities.

Nowadays, Internet users and lifelong learners can take advantage of services like iGoogle or Netvibes to create a personal view on information they are interested in. iGoogle and Netvibes are Mash-up Personal Environments that allows their users to add and combine different information sources of the Internet at one place. Further, by integrating Web2.0 services like Blogs, Delicious or Slideshare the user can follow other users and integrate social networks into such a Mash-up Personal Environment. Consequently the concept of Mash-Up Personal Learning Environments is also applied to support learning. They are called Personal Learning Environments (PLEs) [1] or Mash-Up Personal Learning Environments (MUPPLEs) [2].
MUPPLEs initially support informal learning as they require no institutional background, curriculum structure, and are free of use. Their focus is on the learner independently from institutional needs like student management or assessments. Although, they are most appropriate for informal learning, educational scenarios are thinkable where MUPPLEs become integrated into formal learning like in universities as well. In an even more complex manner the existing of MUPPLEs force universities to open up their environments and services to MUPPLEs [3].

The possibilities of the Web2.0 tools and Mash-Up Personal Environments increase the amount of information available on the Internet and enables people to access almost anything they need. On the other hand, the Internet opens the door for a plethora of information that makes it difficult to get an overview and to select the most suitable information. This selection problem also applies to lifelong learners who get lost on the Internet. The learners can be overwhelmed by the information they receive or they might have problems selecting the most suitable learning content for their personal competence development.

Promising technologies to support people, in order to navigate to the most suitable information, are recommender systems. They are successfully applied at e-commerce web sites like Amazon.com, where people receive recommendations based on the products they are interested in. The recommender system matches customers with a similar taste of products and creates a kind ‘neighborhood’ of like-minded customers. It looks for related products purchased by the neighbors and recommends these to the current customer. This navigational support by recommender systems may help us to reduce time and costs involved in selecting suitable information on the Internet. It will help learners in selecting learning content according to their individual needs, preferences and learning goals.

This article presents the ReMashed system, a MUPPLE that exists out of mashed information from Web2.0 services and a recommender system that offers advice to find most suitable learning content for individual competence development of lifelong learners. The ReMashed system was initially designed to offer navigational support to lifelong learners in informal learning settings. In this article we want to discuss its ability to be used also in formal learning settings.

In the following section we present first related work on MUPPLEs (section two). Afterwards we present the ReMashed system in version 1.0 (section three). After that we present two recommendation approaches for using ReMashed either in formal or informal learning settings (section four) and finally discuss future research (section five).

2 Related work

Nowadays, ‘mashing’ information becomes a widely used activity on the Internet. In order to get an overview of conducted research on MUPPLEs we created a Mindmap that shows related research topics around MUPPLEs

1 The Mindmap can be found at http://www.mindmeister.com/15237440/r-d-on-muppies. Feel free to join us and further extend it.
We identified three main research fields on MUPPLEs: 1. Pedagogical scenarios, 2. Use case studies, and 3. Technology Development. Surprisingly, research field 3 – Technology Development contains most of all published articles. Because of the size of this article we can hardly discuss all related research efforts. Nevertheless, we want to give some examples for each research field and recommend to have a look on the online version of the Mindmap.

According to research field 1, there are multiple pedagogical scenarios thinkable a good example for using a MUPPLE in formal learning can be found in [3]. They present an approach how universities can take advantage of MUPPLEs. An example how MUPPLEs support informal learning can be found in [4]. For research field 2 – Use case studies, we have to point to [5] where four experiments with MUPPLEs are reported. According to research field 3 – Technology Development, various efforts are published. For instance [6, 7] have different approaches to make the development of MUPPLEs more end-user friendly, further [8, 9] try to improve the information interoperability between widgets in MUPPLEs.

The ReMashed system belongs to research field 3 – Technology Development of MUPPLEs and extends the already existing developing efforts with recommender system research. It offers navigation support for lifelong learners to meet most suitable learning content in Mash-Ups. The combination of different Web2.0 services to recommend information based on mashed tag and rating data was not done so far and especially not for learners in MUPPLEs. Thus, ReMashed offers a new approach by mashing data of users from various Web2.0 services to provide tailored recommendation to them.

3 The ReMashed System

In order to test different recommendation approaches for MUPPLEs we designed the ReMashed environment that enables lifelong learners to add their Web2.0 sources to the ReMashed community. The system allows the learners to personalize emerging information of a community to their preferences. They can rate information of the Web2.0 sources in order to define which contributions of other members they like and do not like. ReMashed takes the preferences into account to offer tailored recommendation to the learner. It uses collaborative filtering [10] to generate recommendations by matching together learners with similar opinions about learning resources. Each member of the system has a ‘neighborhood’ of other like-minded learners. Ratings and tags from these neighbors are used to create personalized recommendations for the current learner. The recommender system combines tag and rating based collaborative filtering algorithms in a recommendation strategy. Such a recommendation strategy reacts on certain situations by using the most suitable recommendation technique. We tested a prototypical version of the system (Version 0.9) in an usability evaluation in a group of 49 users from 8 different countries [12]. The evaluation phase ran for one month and was concluded with an online recall questionnaire. In that timeframe 4961 contents were collected, 420 resources were rated and 813 recommendations were offered. The overall satisfaction with the system was positive. The participants suggested particular improvements we took into
account for ReMashed release 1.0 (see Fig. 1). The goals of ReMashed and a detailed system architecture can be found in [11]. Currently, Version1.0 supports the Web2.0 services Twitter, flickr, RSS feeds from Blogs, slideshare, delicious, and YouTube.

Fig. 1. The user interface of the ReMashed 1.0. On the left side, the mashed information from delicious, blogs, flickr and slideshare are shown. On the right side, the recommendations for the current learner are presented.

In order to support the learners with suitable recommendations for learning we extended the user profile with a learning goal specification and related knowledge level definition. The learners can specify three main learning goals with their current knowledge level. They can specify their knowledge levels in a self-assessed way (see Fig. 2).

We supported the learning goal interface by a simple auto-suggest / auto-completion algorithm [13] which can be fed by different technologies like ontologies, metadata, existing tags, or already entered learning goals of other learners.
Initially ReMashed was designed to support lifelong learners in informal learning processes but as Web2.0 is also an important topic for companies and universities we want to explore the application of ReMashed also for formal learning settings. Therefore, ReMashed can be further developed for information dissemination and managing tasks within organizations. Because organizations often have classified information more restricted tools for information sharing are needed. Free accessible Web2.0 services that are open to the Internet can hardly be applied for this purpose. Thus, a kind of closed IntraWeb2.0 solutions (own blogging or twitter system) could be used within the organizations to strengthen the knowledge dissemination. The produced content could emerge in similar interfaces like the suggested ReMashed system, without fearing to lose classified information to third parties. In addition, companies or universities might have more restricted learning goals than individual lifelong learners. Therefore, we have to adjust the recommendation approach to certain conditions of formal and informal learning settings.

In the following section we discuss the changes in recommendation approaches to support formal and informal learning settings with ReMashed.

4 Recommendation approaches for formal and informal learning settings

Different recommendation needs can be identified when considering the use of Technology-Enhanced Learning (TEL) to support different educational settings like formal and informal learning.

Formal education, being usually organized according to some curriculum, traditionally occurs in teacher-directed environments with person-to-person interactions. Informal learning is described in literature as a learning phase of so called lifelong learners who are not participating in any formal learning context (like universities or schools). Lifelong learners are acting much more self-directed and they are responsible for their own learning pace and path [14]. In addition, the learning content for their learning might come from many different sources: expert communities, work context, training or even friends might offer an opportunity for an informal competence development. The learning process is also not designed by an institution or responsible teachers like in formal learning, but it depends to a large extent on individual preferences learners have or choices that learners take.

Depending on the learning settings, the aims of TEL systems, their environmental conditions, and the tasks that they support also change. Thus, considering the way TEL context variables vary according to the adopted setting, the information needs of the targeted users change. This can greatly affect the design of recommender systems for different TEL settings. In the following sub section we suggest recommendation approaches for formal and informal learning settings. We especially focus thereby on the learning goal interface and its related knowledge levels in ReMashed. Both approaches can feed the learning goal interface (see Fig. 2) to embed pedagogical reasoning into the recommendation.
4.1 Recommendation approach for formal learning settings

Many recommender systems for formal learning like [15] using fine granulated knowledge domains and can therefore offer personalized recommendations to the learners. They can use metadata and ontologies to define the relationships, conditions, and dependencies of learning resources and learner models. They can take advantage of well structured formal relationships like predefined learning plans (curriculum) with locations, student/teacher profiles, and accreditation procedures to recommend courses or personalize learning.

A promising solution to apply the learning goal functionality of ReMashed for formal learning settings is the work on adaptive sequencing [16]. It takes into account individual characteristics and preferences for sequencing learning content. Adaptive sequencing needs three abstraction layers to add semantics between content and possible learning goals. On the top layer, the learning goal layer, a hierarchy of learning goals needs to be modeled. In the second layer, the conceptual layer, ontology of domain concepts needs to be created. The lowest layer, the content layer, contains the actual content of the learners. With the adaptive sequencing approach organizations can pre-structure and control the available learning goals within the system. Further, they can easily adjust and cluster the learning goals and their related knowledge level to their needs.

This approach requires a maintenance effort as there are many design activities needed before the runtime and also during the maintenance of the system. In addition, the knowledge domains in the learning environment need to be described in detail. These aspects make the adaptive sequencing approach not applicable for informal learning settings.

4.2 Recommendation approach for informal learning settings

In order to apply ReMashed for informal learning settings, we need different recommendation approaches. The absence of maintenance and structure in informal learning settings is also called the ‘open corpus problem’. The open corpus problem applies when an unlimited set of documents is given that cannot be manually structured and indexed with domain concepts and metadata [17]. For instance, learning goals in informal learning settings are a rather diffuse parameter because they rely on information given by the learners without any standardization.

To handle the dynamic and diffuse characteristic of learning goals and related knowledge levels probabilistic techniques like text driven latent semantic analysis or mining techniques like hierarchical clustering are promising. They are appropriate for informal learning settings because they require nearly no maintenance and improve through the emergent behaviour of the community. ReMashed is not a purely text based system, as it uses videos and pictures as well. Therefore a purely text-driven approach is rather inappropriate. Most suitable therefore is the hierarchical clustering method. It builds up a hierarchy of items by continuously merging the two most similar items / groups into a new group. In the ReMashed case, an item is a single blog posting or a picture with related tags. We can create a measure for most
frequently used keywords for each blog or picture by simply using words counts. The similarity between items / groups can then be measured by similarity measures like Euclidean distance or the Pearson correlation. In each iteration the method calculates the distance between every pair of items / groups and the closet ones are merged together to form a new group. This process is repeated until there is one group with various sub groups. In that way hierarchical clustering creates a kind of automated ontology on top of the available contents.

Similar like the adaptive sequencing approach this automated ontology can be fed into the learning goal interface in ReMashed and present the available learning goals in the system.

5 Conclusions and Future Research

This article presented the ReMashed system, an MUPPLE with a recommender system for learners. The article presented related work in MUPPLE in section two. Further it presented the ReMashed system in section three and discusses its application for formal and informal learning settings by using different recommendation approaches (section four).

Based on formal or informal learning setting of learners recommendations have to be done rather differently. In case ReMashed is applied in formal learning settings the recommender system can take advantage of well structured formal relationships like predefined learning plans (curriculum) with locations, student/teacher profiles, and accreditation procedures to recommend learning resources to learners. Therefore, more top-down technologies like ontologies and adaptive sequencing are suitable to offer personalized recommendations.

In order to apply ReMashed for informal learning settings, we need to take into account a lack of structure and maintenance. Thus, to handle the dynamic and diffuse characteristic of informal learning more bottom-up technologies like latent semantic analysis or mining techniques like hierarchical clustering are needed. They are more appropriate for informal learning settings because they require nearly no maintenance and improve through the emerging behaviour of the community.

In the future we want to explore most suitable recommendations approaches for the use of MUPPLEs in formal and informal learning. Furthermore, we want to improve interoperability of ReMashed to other MUPPLEs. Therefore, ReMashed has to provide a widget interface to enable lifelong learners to integrate the recommendations from ReMashed into their MUPPLEs. Such a widget should provide recommendations and the possibility to rate content from the community to further personalize the needs of the learners.

Acknowledgement

Authors’ efforts were (partly) funded by the European Commission in TENCompetence (IST-2004-02787) (http://www.tencompetence.org).
References


3. Hermans, H., Verjans, S.: Developing a sustainable, student centred VLE: the OUNL case. 23rd ICDE World Conference on Open Learning and Distance Education 2009 EADTU Annual Conference (M-2009), Maastricht, The Netherlands (2009)


Mapping Web Personal Learning Environments

Matthias Palmér¹, Stéphane Sire², Evgeny Bogdanov², Denis Gillet² and Fridolin Wild³

¹ Royal Institute of Technology (KTH), Stockholm, Sweden, matthias@nada.kth.se
² École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, {stephane.sire, evgeny.bogdanov, denis.gillet}@epfl.ch
³ The Open University, Milton Keynes, United Kingdom, f.wild@open.ac.uk

Abstract. A recent trend in web development is to build platforms which are carefully designed to host a plurality of software components (sometimes called widgets or plugins) which can be organized or combined (mashed-up) at user’s convenience to create personalized environments. The same holds true for the web development of educational applications. The degree of personalization can depend on the role of users such as in traditional virtual learning environment, where the components are chosen by a teacher in the context of a course. Or, it can be more opened as in a so-called personalized learning environment (PLE). It now exists a wide array of available web platforms exhibiting different functionalities but all built on the same concept of aggregating components together to support different tasks and scenarios. There is now an overlap between the development of PLE and the more generic developments in web 2.0 applications such as social network sites. This article shows that 6 more or less independent dimensions allow to map the functionalities of these platforms: the screen dimension maps the visual integration, the data dimension maps the portability of data, the temporal dimension maps the coupling between participants, the social dimension maps the grouping of users, the activity dimension maps the structuring of end users’ interactions with the environment, and the runtime dimension maps the flexibility in accessing the system from different end points. Finally these dimensions are used to compare 6 familiar Web platforms which could potentially be used in the construction of a PLE.

Keywords: widget, mashup, web, comparison, PLE.

1 Introduction

One of the key outcomes of learning is the construction of a learning environment, i.e. that set of tools that brings together people and content artefacts in learning activities to support them in constructing and processing information and knowledge. These environments are distributed and networked by nature. When looking at personal arrangements in this learning ecosystem, i.e. an individual’s selection of tightly-
loosely-coupled tools, close and distant contacts, both created and consumed objects, used for and in main as well as side activities, we speak of a personal learning environment (Wild, 2008).

Looking at the digital parts of this environment, a rich set of implementation approaches can be found in previous work on personal learning environments. Early work (e.g. Liber, 2000; Kearney et al., 2005) focuses mainly on conceptual issues, the next phase is characterised by an emphasis on interoperability issues (Downes, 2005; Wilson, 2005; Wilson et al., 2007) and a stronger emphasis of linking personal learning environments to social software: data interoperability, most notably RSS/ATOM-based aggregation, and service integration of web-services such as the storage/retrieval services offered by the Flickr API.

Whereas in recent advances Wilson et al. (2007) propose to differentiate implementation strategies into coordinated use, simple connectors for data exchange and service interoperability, and abstracted and generalized connectors in form of conduits, we propose within this contribution to further differentiate the latter two into a set of six dimensions with corresponding implementation features. These six PLE dimensions encompass screen, data, temporal, social, activity, and run-time. At the end of the paper we use these dimensions and features to compare six different platforms. We believe the result can be used both to make decisions on which platform to use today as well as identify the trends and areas where further investigation are needed to pave the way for better PLEs.

In our subsequent elaboration on these dimensions of a web PLE architecture, we will be using the following specific terminology. First of all, the software components sometimes referred to as applications, plugins or widgets that are hosted in the PLE will be referred to as simply widgets. Second, we will refer to the server side code that manages the main functionalities of the PLE as the PLE engine. Third, the settings used to initialize and deploy the hosted widgets will be referred to as widget preferences and taken together as the PLE configuration which might be stored locally in the PLE engine or remotely via a dedicated configuration service. Fourth, the web container which renders and executes the widgets on the client-side together with common facilities such as navigation between the widgets will be referred to as the PLE container. Finally, PLEs may provide a way to organize sets of widgets together, for instance into tabs, which can constitute a learning context, sometimes with support for collaboration. We will refer to such sets of widgets as PLE spaces.

2 Overview of Dimensions

The Table 1 summarizes six dimensions of functionality to be used when measuring the PLE characteristics of web platforms. The definitions of the dimensions are made to both capture as many relevant features as possible as well as to make them independent, implying that web platforms can support any combination of them.

These dimensions are most of the time independent and they are not all necessary to build a PLE. For instance a user that would select a Netvibes personalized homepage as their PLE would more or less only use the screen dimension. It is even
conceivable that somebody uses a PLE with none of these dimensions. This is the case for a user that select a simple blogging tool such as Wordpress to self-reflect on her learning process by writing text snippets (without plugins and comments otherwise there would be some elements of the screen and social dimensions).

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Definition</th>
<th>Potential Standards</th>
</tr>
</thead>
</table>
| Screen    | Organization of several widgets within a PLE in a spatial manner. | W3C Widgets 1.0  
Google Gadget API  
Google GadgetTabML  
Netvibes UWA  
OpenAjax Metadata 1.0 |
| Data      | Interoperability of data and metadata across widgets and underlying services  
Includes issues with cut & paste, drag & drop, data formats, protocols, semantics. | Various Data and metadata standards such as RSS, SCORM, Dublin Core RDF, HTML5 Dn'D  
OpenAjax Hub 2.0  
Google gadget pubsub |
| Temporal  | Updates to widget configuration, state or data is more or less synchronous with other active users who share the widget instance. | COMET/Reverse Ajax  
XMPP, XMPP over BOSH  
Google Wave Federation Protocol |
| Social    | Interoperability of user identity, profile information and list of friends.  
Ability to define some group contexts for sharing widget events, state or data. | OpenID (portable profiles)  
OpenSocial API  
Portable Contacts  
Facebook Connect (Friend Linking and Social Distribution)  
FOAF |
| Activity  | The applications in use in the PLE can be controlled through scripts that engage the user into learning activities. | WS-BPEL (business oriented)  
IMS Learning Design Specification (targeted at VLE and design-time oriented, not run-time) |
| Runtime   | Cross PLE interoperability allows you to exchange one rendering and execution platform or its parts with another. | W3C Widgets 1.0: Packaging and Configuration  
OPML  
Open Ajax Mashup  
Reference Application |

However, we believe that the power of web PLEs will be to support several of these dimensions. This will allow more powerful widgets, such as for instance
collaborative widgets by including elements of the screen, temporal and social dimensions. But it will also make the web PLEs more reliable environments, for instance the data dimension will make your data more portable to avoid data lock in. Furthermore, the runtime dimension will allow you to switch to another PLE of your choice with minimal migration issues and the feature of being able to collaborate across different PLEs will allow you to stay even when your collaborators are using different PLEs.

The diagram below shows these dimensions graphically in what could be an abstract view of a generic PLE. In this diagram we have made explicit that we see the PLE as an Integrated Development Environment (or IDE), this is because it allows you to develop a learning environment to fit your needs rather than force you to be satisfied by what is given. For instance in a PLE with social integration, a part of the user interface is dedicated to invite friends and to accept invitations. Similarly in a web personal Home page a part of the user interface is dedicated to browse widgets and to place new widgets on a grid on the screen.

![Diagram showing dimensions](image)

**Fig 1.** The dimensions for building web Personal Learning Environments.

In the next sections, we examine each dimension in greater detail.

### 3 Screen Dimension

Many platforms allow multiple distinct software components to exist side by side within a container application. Examples range from simple pasting of html-snippets inside blog posts to advanced standard based widget containers or even web Desktops that comes with a full set of tools. Below we outline a few possible features of a container application with respect to the screen dimension.

**Shared screen.** Many Web sites, such as video sharing sites, allow their users to cut and paste short snippets of (X)HTML code that allow to project their content. These
snippets typically uses either an `<iframe>` tag to embed a separate web page / application, `<object>` tag for embedding of flash or java applets, a `<script>` tag for loading a separate javascript that dynamically builds up the user interface, or any combination of the above. For blogging or wiki platforms (e.g. WordPress, MediaWiki) this approach is a good alternative. For Content and Learning Management Systems it is more common with plugin based approaches. In general, plugins require more administration or even programming skills to be made available. Depending on the level of automation of the system you might need access to the underlying installation to reconfigure or activate new plugins.

**Widget standards.** The most limiting factor of plugins are that they are by definition limited to a single platform. If a plugin infrastructure grows beyond the boundaries of a single platform and brings with it a specification and good documentation it is perhaps more reasonable to talk of a de facto standard, lets call it a widget standard in this case.

Google gadgets are a typical example of a de facto standard while for example Netvibes UWA, Wordpress plugins and Facebook applications are somewhere in between. In addition there are standardization initiatives like the W3C widget 1.0 specification (W3C, 2009b), the OpenAjax metadata specification (OpenAjax Alliance, 2009b) or the BONDI specification (Open Mobile Terminal Platform forum, 2009). The benefit of adopting widget standards are that they help to establish an ecosystem of widgets that can run on many platforms, for example Google gadgets can run both in Wordpress, Netvibes and Facebook.

**Layout of widgets.** Platforms that have a clear focus on a single activity, like writing blog posts or editing wikipages will probably not have more than a few widgets active at the same time. However, with more advanced PLEs it must be possible to focus on different kind of activities which implies a wider range of supportive widgets being accessible. Hence, the need for choosing your own layout and order the widgets according to your own needs increases. A common approach is to allow widgets to be placed on multiple dashboards that are available as tabs.

**Web Desktop.** To lower the barriers for non advanced users a truly capable PLE probably need to provide a default set of generic tools similar to a regular pc desktop. Even though these tools are widgets as well they are likely to be available from a menu, to minimize clutter, rather than laid out on a gigantic dashboard by default. This brings us close to the concept of Web Desktop or Web Operating System which are web applications that mimics the desktop, examples include EyeOS and G.ho.st. There are several aspects of the regular Desktop experience that could be brought along, for example the idea of sessions where widgets can be opened or closed and access to content (independent of widget) via something like a file manager.
4 Data Dimension

There are many forms of data portability which have been described elsewhere (Turnitsa, 2005). In web 2.0 data portability is often seen as data created or stored within application A can be read and/or copied/moved into application B and interpreted by application B, see for example the vision statement from the DataPortability project. What we see today is mostly simple widgets using established standards such as RSS while more advanced widgets introduce their own the choice of data formats and services for content and own vocabularies for preferences. The latter choices are nearly always made based on the perspective of a individual widgets and not a wider widget landscape. The result are widgets that have little or no knowledge of other widgets or even the surrounding context and the data is often specific for a single widget. More recently we have seen the emergence of client-side communication protocols that allow applications integrated at the presentation level (screen dimension) to exchange data on user's behalf (autonomously or with methods such as Dn'D).

A web PLE could provide good supporting structures that both enable widget developers to provide more cross widget features as well as help them to make good decisions on data portability. Below we identify four such supporting structures, features, that a web PLE could provide with respect to the data dimension.

- **Inter-widgets communication.** Some recent containers have started to propose client-side data portability, that is the ability for applications to exchange data directly within a host container without interfering with their server-side implementation, see (Sire & al., 2009). Such features can be supported with the integration of a communication layer within the host container, examples include Google gadget "pubsub" (Google, 2009a) and the OpenAjax Hub Publish/Subscribe APIs (OpenAjax Alliance, 2009a).

- **Drag and drop.** With the upcoming HTML5 (W3C, 2009a), drag and drop will be available natively in the browser. But it is also possible to achieve similar effects already today via a Javascript communication layer in the client.

- **PLE data manager.** An increasingly common approach to data integration consists in providing storage services apart from the applications that need storage. The storage services allow several applications hosted on a container, or several instances of the same application hosted on the same engine, to share a common data store. This service can be part of the PLE, such as in OpenSocial containers with the data persistence API, or they can be offered by independent providers such as with the Google Wave Federation Architecture. A PLE data manager could also start by providing support for a few common protocols such as Atom Publishing Protocol and search APIs (e.g. OpenSearch, SQI) to allow the development of unified search services into a PLE.
Linked data support. The main difficulty of client-side data portability is to determine the compatibility of two components before establishing a communication channel between them. Typically they require the agreement on common microformats to exchange data (e.g. hCalendar, FOAF) or meta-data about the data (e.g. SCORM, Dublin Core). A semantic driven approach using basic HTTP with common RDF properties (linked data) may offer greater flexibility and scalability as well as enable both interoperability on the fly and automatic data mash-ups. As the world will continue to contain non-semantic data sources in the foreseeable future a semantic gateway will be needed. SA-REST (Lathem & al., 2007), GRDDL, RDFa and similar initiatives could be really useful for the realization of such a gateway.

5 Temporal Dimension

Collaborative aspect of web PLEs brought a new dimension to people. It is not enough to only assign users the specific rights over a particular shared object, but it is equally important to notify users about current content and changes in this object. In other words, the state changes in a shared object should be propagated to all people engaged.

The initial architecture of the Internet allowed update to object state to happen only on page reload. Thus, if two people worked on the same document, the first person could see the changes of the second person only after clicking the refresh button in the browser. Initial workarounds such as polling was unsophisticated but recently, the situation has dramatically improved. With the advent of online chat applications, new protocols were invented (COMET, Reverse Ajax, XMPP, etc.) which allowed updates to be synchronous propagated. With these new techniques in mind we outline four features that a PLE can support regarding the temporal dimension.

Push data updates. A common approach to synchronous updates is to have a Javascript API that allows widgets to push data to other instances of the same widget within a well defined context. All users that have that widget in the same context will see the update without reload. The Wookie engine provides exactly this functionality as an extension to the Widget 1.0 APIs and Events specification. The Google realtime gadgets API is another initiative that today allows widgets instances running inside a Google online chat to exchange data although it has been stated that it will soon work on both the iGoogle and Orkut platforms (Google, 2009b).

Push preference updates. Many simple widgets have no data (except static resources) except preferences. A typical example is a map widget that shows a specific location. A user who wants to show a specific position in the map widget should be able to do so without forcing all other users sharing the context to reload their PLEs. A more advanced example could be that the PLE space must change to reflect a new need, for example add and configure a new widget.
Real time data updates. The distinction between push and real time data updates are important for two reasons. First it puts higher demands on the infrastructure since delays are more clearly noticed. Second depending on the character of the data, data updates might need to be merged and locking mechanisms to avoid conflicts is not an option due to the real time aspect. For example, if two people edit in the same place in a collaborative text document there is a risk for conflicts but if you have a voting application the votes should be stored separately, making conflicts impossible. Google Wave is an example of a platform that provides real time updates.

Data and preferences history. With data updates coming from different users the need to be able to go back to view or even revert to an older version of the history becomes more important. Optimally the history should include both the widget preferences, the wider PLE configuration as well as other data. Examples of systems that include history today are wikis, content management systems, and Google Wave via it's playback mechanism. Google Wave playback is especially interesting since the versions are not necessarily distinct in the traditional document centric perspective, instead there is a continuum of real time updates that can be scrolled through.

6 Social Dimension

Web 2.0 applications explicitly model the concept of the user and of her list of friends, or followers. This information is used to exploit the social graph of the user when sharing application data and posting notifications. Generally speaking, the social graph information is used by the social container to dynamically define different groups when a user is interacting with an embedded application, either on his own page, most of the time called the user's profile page, or on the pages of other users. We have identified four features that a web PLE can support in the social dimension.

List of Friends. This is the most basic level where a system supports users to have a list of friends. Many systems requires a step of confirmation before friendship is established and sometimes the friendship is qualified into more specific relation categories such as friends, relatives, colleagues, etc. To avoid the hassle of recreating your list of friends manually, many systems provide import/export mechanisms supporting formats such as vCard or FOAF.

Friends server. Unless users are very organized, they often run into synchronization issues when importing/exporting their list of friends between systems. Another approach, more modern and effective, is to have his list of friends in a single system and make them appear in other systems via a protocol such as Facebook connect or OpenSocial.
Access control. The OpenSocial protocol defines several groups such as owner, viewer, friends of owner, friends of viewer (OpenSocial Foundation, 2009b). These groups are used to define access control policies to application data, and to define data storage mechanisms that allow users to send data to their friends, or to consult data from their friends, applying complex rules for developers. For instance a common rule to facilitate the viral spreading of applications within social networks is that it is usually required to have installed the application on her own profile before being able to post data through this application on a friend's profile. Some rules are also used to govern reading access to the activity streams of a user and to manage writing of notifications from a user into other user's news feeds (Facebook Developers, 2009). The goal of these rules is to leverage the social graph of the user for making decision and proposing recommendations (OpenSocial Foundation, 2009a).

Independent groups. We can easily see how the social graph dimension could also leverage the educative power of a PLE by allowing to support learning activities in groups. However our impression is that currently the flexibility to create and manage user's profile which has been gained with protocols such as OpenId is still missing for managing group's profile in a way that is independent of the PLE engine. More flexibility will be needed to define different groups which may be related to different user's activities and to be able to import these groups into different PLE engines depending on the context. One approach to solve this could be to do an extension to OpenId, perhaps called GroupId.

7 Activity Dimension

Activity theory states that an activity is shaped by its surroundings (Leont'ev, 1947; Engeström, 1987). For example, tools do have certain affordances: a door knob lends itself to opening. On the contrary, activities do also shape their surroundings: they can result in the construction of a tool. Below we identify four features that provides various levels of support for activities in the surrounding Web PLE.

Manual guide. The simplest approach for guiding the user through a series of actions is simply to have instructions easily available in the environment. Simple checkboxes or textareas where the user can mark and reflect on progress are a nice addition. If there are decision points a more complex view is needed where branches can be opened or closed, again based on manual input. Even though a manual guide is achievable in a regular widget, giving it a more prominent position in the environment might enhance the feeling of guidance for the user.

Flow enables widgets. Based upon a script and manual input of the user, certain widgets can become enabled or visible, leading the way for the user. Some care has to be taken since when enabling web applications involved in learning activities to become part of scripted activities (Dillenbourg & Jerman, 2007), matters of
orchestration of how users should use a set of tools and services to achieve a certain learning goal demand attention.

**Scripted inter-widgets data flow.** In addition to simply activating or enabling widgets based on manual input you can let scripts do it for you, based on the resulting produced/transformed/aggregated data from other activities. Furthermore, the widgets preferences or even starting data can be initialized from these earlier activities. The difference to the inter-widgets communication feature in the data dimension is that the data flow is driven by the script, not the individual widgets. The script might even take into account the social dimension, making use of different users contribution. SCORM runtime and Learning Design are two example of standards that try to encode the scripting logics in combination with content rather than widgets. Mupple (Wild et al., 200) is one example of a activity scripting environment in a widget context. The underlying learner interaction scripting language (LISL) allows to specify action/outcome/tool bindings which can render the user interfaces of web applications into one common dashboard like environment thereby providing a kind of human self-instructing and scaffolding into defining intended outcomes.

**Recommendations.** A more advanced case is when the decisions, data flow and recommendations are performed by a independent software agent. It might, for instance, consult remote services, investigate user history, collect attention metadata, know about desired outcomes and aspired goals, and then make inferences to provide guidance in the form of recommendations. There is many things that can be done on this level, for instance (Kildare et al., 2009) deals with explicating interaction norms and rules that can thereby be monitored automatically, thus relieving the user of certain regulation work within collaborative scenarios. In this case, rules are formulated in statements for the Java Expert System Shell (JESS).

8 **Runtime Dimension**

In the future we do not expect that there will be a single web PLE that everyone uses. Instead, we find it more realistic that there will be many distinct web PLEs that compete with slightly different functionality, design and interaction paradigms. Furthermore, we believe that the full strength of having a range of alternative web PLEs is not reached until it is both easy to move between PLEs and also to collaborate across PLE boundaries. Thus, we believe it is crucial that a widget or an entire PLE space in one PLE can be easily experienced in another PLE. Below we list four features that a PLE can support with respect to the runtime dimension.

**Feed export and import.** Feeds are today the bread and butter for most widgets, and many PLE platforms provide specific support, for instance, allowing you to add a feed directly rather than adding and configuring a widget to display it. It is also common to provide import and export of OPML files (listing feeds with feed names, possibly in
groups) allowing simple migration to other PLE environments or dedicated feed readers.

**Generic export and import.** Even though feeds are important there are many widgets that use other forms of data that will not survive the transfer via an OPML file. Providing a lossless export and import of a PLE configuration from the same system is the first step, iGoogle provides GadgetTabML that does this. As the second step, the format needs to be widely accepted, hopefully even standardized, to allow lossless (or close to it) export and import between PLEs. Although, even if such a lossless export and import is supported between PLEs, it depends on both widget standards as well as authentication and authorization protocols to be in place already.

**External configuration.** By keeping the PLE configuration in a separate service distinct from the PLE engine we accomplish three things. First, users need not import or export, they will have a single reliable access point to point their PLE to. Hence they could easily switch between or even use multiple PLEs in parallel to get the best out of each service. Second, following each others PLE spaces, or even collaborating, will be possible across PLEs. Typically a teacher could invite students into courses that they could partake in a PLE of their own choice. Third, keeping PLE configurations in a dedicated service will encourage additional levels of functionality such as a PLE configuration ecosystem where the use of templates could help users to get started and teachers to establish good practices.

**Embedding.** In the screen dimension section we discussed different alternatives to how software components, such as widgets, can appear together in a PLE, but we did not discuss how an entire PLE could be embedded into another environment. The approach to paste HTML snippets that via iframe, embed or script tags is viable here as well. A simple example that people are using is running Facebook as a tab in iGoogle via the canvas view (a full screen iframe widget). Clearly iframe embedding is rather weak from an integration perspective. When the host environment is a PLE it would be beneficial to have a deeper integration, especially along the social and data dimensions outlined above. Such a deeper integration would rely on Javascript APIs and Javascript wrappers probably in combination with a few standardized protocols. One possible example of this is the Google Wave embed API, however it is still unclear if Google Waves should be compared with an entire PLE or a very advanced widget. The Open Ajax Mashup Reference Application proposes a portability solution to individual widgets, as does the Widget Server in the Wookie widget engine. This could be generalized beyond single widget rendering to support full PLE configuration rendering.
9 Mapping Existing Web PLEs

In this section we will map six different platforms with respect to the six PLE dimensions. The platforms we map are the personal/social start pages iGoogle (IG) and Netvibes (NV), the Learning Management System Moodle combined with the Wookie engine (M&W), the collaboration platform Google Wave (GW), the mashup platform Afrous (AF), and the web desktop G.ho.st (GH).

We have investigated the support for each of the four features per dimension (introduced above) and simplified the findings to whether it is supported or not. The investigation has been done by reading available documentation and testing the platforms (except Google Wave as it is not available for the general public yet). As the features are not absolute there are situations when there is only partial support (reported as supported) or that the feature is not applicable (then reported as not supported). A full explanation for why we have deemed that a platform has support or not for a specific feature is unfortunately out of scope of this paper although some cases are more or less explicit from the discussion in the sections above.

<table>
<thead>
<tr>
<th>Abr.</th>
<th>Platform</th>
<th>Type</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>IG</td>
<td>iGoogle</td>
<td>start page</td>
<td><a href="http://www.igoogle.com">www.igoogle.com</a></td>
</tr>
<tr>
<td>NV</td>
<td>Netvibes</td>
<td>start page</td>
<td><a href="http://www.netvibes.com">www.netvibes.com</a></td>
</tr>
<tr>
<td>M&amp;W</td>
<td>Moodle + Wookie</td>
<td>LMS</td>
<td>getwookie.org/moodle/</td>
</tr>
<tr>
<td>GW</td>
<td>Google Wave</td>
<td>communication and collaboration</td>
<td>wave.google.com/help/wave/about.html</td>
</tr>
<tr>
<td>AF</td>
<td>Afrous</td>
<td>mashup</td>
<td><a href="http://www.afrous.com">www.afrous.com</a></td>
</tr>
<tr>
<td>GH</td>
<td>G.ho.st</td>
<td>web desktop</td>
<td>g.ho.st</td>
</tr>
</tbody>
</table>

It is important to notice that the results should not be seen as a total evaluation of these platforms as they might have other features not evaluated here. For example, Afrous seems to be a very capable client side mashup platform and G.ho.st a good web desktop, however, this does not necessarily make them good web PLEs as defined by our dimensions.
**Fig 2.** Comparing the features of the 6 platforms mapped onto 6 dimensions.

**Table 3.** Feature level details of the comparison of the 6 platforms.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Feature</th>
<th>IG</th>
<th>NV</th>
<th>M&amp;W</th>
<th>GW</th>
<th>AF</th>
<th>GH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space</td>
<td>Shared screen</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Widget standard</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Layout of widgets</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Web desktop</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>Inter-widgets communication</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Drag and drop</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PLE data manager</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Linked data support</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temporal</td>
<td>Push data updates</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Push preference updates</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Real time data updates</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Data and preference history</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Social</td>
<td>List of friends</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Friends server</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Independent groups</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity</td>
<td>Manual guide</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
10 Conclusion

The Future of web PLE relies on the integration of several dimensions which have been historically developed separately for web applications such as widget portals, social networks and content management systems, and which are now converging. This allows new scenarios with more personalization and more group usage of web applications. However this approach will be successful and support the growth of web PLE usage in education only if we can see a convergence between the multiple standardization efforts which are now underway at the World Wide Web consortium, Open Ajax alliance, Open Mobile Terminal Platform forum (OMTP), DataPortability project, at big companies such as Facebook and Google, the OpenSocial foundation, etc.

In this paper we have identified 24 features along six dimensions that can be used to map the functionality of web PLEs today. We have also investigated six platforms and presented the results as spider diagrams. If conclusions of the results are to be drawn, it can be noted that Google Wave and Moodle & Wookie are the only PLEs that are strong in the temporal, social and activity dimensions indicating a strong focus on collaboration. While the other platforms, iGoogle, Netvibes, Afrous, and G.ho.st have their focus on the data, screen, and runtime dimensions indicating a higher focus on personal customization. As we believe that good PLEs will need both strong personal customization and collaboration features, the conclusion is that there is still a lot of room for future web PLEs to realize that potential.

It should also be noted that Google Wave is at the time of writing a very young platform that we yet do not know very much about, it is very likely that it quite soon will score much higher on the data and runtime dimensions. However, that it will score higher in the screen dimension is not so obvious to us. If an individual wave is just a conversation or more what we have referred to as a PLE space, space remains to be seen.

The weakness of all but the Moodle & Wookie platform in the activity dimension may indicate that it should not be seen as a separate dimension of a PLE but as features of individual widgets. But, it could also mean that the platforms we have
looked at are yet immature with respect to support for learning activities (which traditional Learning Management Systems excel at). The latter explanation is supported by experience from traditional Learning Management Systems where such features have only been realized with support from the surrounding platform and not being the sole responsibility of individual tools/widgets.

Finally, we hope that these features and dimensions could be further refined and perhaps be useful together with personal or organizational preferences for making decisions of which web PLEs to use. Perhaps can they also provide guidance in future research and development.

Acknowledgement

This work has been co-funded by the European Union under the Information and Communication Technologies (ICT) theme of the 7th Framework Programme.

References

Facebook Developers: Anatomy of an App (2009), online developers.facebook.com/get_started.php?tab=anatomy#news_feed
Google: Gadget-to-Gadget Communication (2009a), online code.google.com/apis/gadgets/docs/pubsu.html
Google: Realtime Gadgets API (2009b), online code.google.com/apis/talk/gadgets_realtime.html
Open Mobile Terminal Platform forum: BONDI Release 1.01 Specification (2009), online bondi.omtp.org/


Wilson, S.: Future VLE (2005), online zope.cetis.ac.uk/members/scott/blogview?entry=20050125170206


Moodle Wave: Reinventing the VLE using Widget technologies

Scott Wilson, Paul Sharples, Dai Griffiths and Kris Popat,
University of Bolton, UK
{s.wilson, p.sharples, d.griffiths, k.popat}@bolton.ac.uk

Abstract. Can widget technologies revolutionize the VLE and challenge the dominant design of elearning? The authors describe an experimental implementation of the Moodle VLE using the W3C Widget specification and Google Wave Gadget API to replace existing core features. This raises four challenges; (1) By replacing most of the functionality of the VLE with simple reusable applications, the core functionality and design of the VLE is challenged; (2) by allowing these applications to be extracted and used in personal learning environments, the hegemony of the institutional system is challenged; (3) by conducting learning activities in small discrete applications, the model of student tracking is challenged; (4) synchronous, live updating applications challenges the model of user behaviour for the VLE.

Keywords: Widgets, Personal Learning Environments, Virtual Learning Environments, Interoperability

1 Introduction

In the paper “Personal Learning Environments: Challenging the dominant design of educational systems”, Wilson et al. [1] describe the VLE (Virtual Learning Environment; also called a Learning Management System) as having become a dominant design for educational systems; this is a situation which limits choices for users and institutions, while also presenting a very challenging environment for an alternative design to gain any traction.

In this paper we describe an approach to challenging the dominant design through creatively subverting the VLE using highly interactive applications (Widgets) that can be delivered within the VLE but also embedded by the users into other platforms, including individually-owned tools and websites. By extending the capabilities of the VLE in this manner, we can create a new conversation about the VLE that moves us away from the dominant design, but stay within the comfort zone of lecturers, managers and students who have become used to the model. Also, rather than attempt to “create” a Personal Learning Environment (PLE) that is provided to learners, we instead open up the VLE to be remixed by users to construct their own PLE using technologies of their choosing.

The approach we took was to extend the Moodle VLE [2] using Wookie Widget Server (both open source), extended to support the Google Wave Gadgets API[3] and to provide embed code to let users take Widgets elsewhere. The experimental system
was made publicly available for testing, and existing Moodle users were encouraged to try it.

2 Architecture & Design

2.1 Design Objectives

The objective for the experiment was to create an instance of Moodle that made use of the emerging W3C and Google Wave technologies in a way that revolutionizes the capabilities and use of the VLE platform, while at the same time requiring little or no modification of the VLE itself. In particular, we wished to deliver a very new kind of user experience based on synchronous, live-updating collaborative applications, and to enable these applications to be easily extracted by users so they could use them in personal platforms, such as Netvibes personal pages.

To realize this design we built on previous work [4, 5] and used Wookie, an open-source Widget engine, to render the Widgets and place them inside Moodle using iFrames (Figure 1). As well as providing a clear separation of concerns – allowing Widgets to be rendered for other platforms – this also provides a measure of security, as provided that Wookie and Moodle are served from different domains, any Widgets are unable to access information from within Moodle by accessing the data on the current page.
2.2 Standards

The key interoperability standards used were the W3C Widgets family of standards, and the Google Wave Gadget API. The W3C Widgets Packaging and Configuration specification [6] defines how Widgets are packaged, and their metadata. W3C Widgets API and Events [7] defines an API for per-instance persistence of preferences for Widgets, and access to metadata attributes. The Google Wave Gadget API [3] defines an API for families of Widget instances that share a common context, with methods to support shared state information and participants.¹

Used together, this package of standards enables authors to create live-updating, collaborative applications that can be packaged and deployed in a standard fashion, with no dependency on a particular server platform or container application.

2.3 Components

The experiment involved the use of a standard Moodle distribution (version 1.9). The customization of this system involved two major components: the Wookie Widget Server and the Wookie Moodle Block.

The Wookie Widget Server is the component responsible for generating instances of Widgets, providing Widget APIs, and managing widget state persistence. For this experiment, the most significant implementation work was updating the Wookie server to also support the Google Wave Gadget API.

¹ The Google Wave platform consists of several different protocols and APIs; the Wave Gadget API is only a small subset of this platform. We did not implement the Wave Federation Protocol or Robots API. This means that, for example, Moodle does not become part of a Wave federation, and courses do not become wavelets; only the mode of interaction with widgets is made available.
The **Wookie Moodle Block** is a small PHP plugin for the Moodle VLE that uses the “blocks” extension mechanism offered by Moodle that allows course administrators to add content to course sidebars. The implementation did not make any changes to the core Moodle VLE at all, only using the Block extension system.

While Moodle is responsible for its usual core functions of managing courses and users, the **Wookie Moodle Block** is responsible for communicating with Wookie to request instances of Widgets, and then delegate rendering of applications to Widgets rendered by **Wookie Widget Server**. Running instances of Widgets then communicate with the **Wookie Widget Server’s AJAX API** and receive push events using its Comet service. This architecture is depicted in Figure 2.

![Figure 2: Detailed architecture](image)

### 3 Implementation

#### 3.1 Setup and configuration

We installed Moodle 1.9, Wookie Server, Tomcat, and MySQL on an Ubuntu server; this server was not dedicated to the experiment, but was instead a shared development server running a number of applications and websites. However, it performed very well under test.

#### 3.2 Implementing the Wave Gadget API

It was relatively simple to map the Wave Gadget API onto our prior implementation of Widget shared state persistence (see [4,5]). We also used the
Moodle Block identifier as a way to generate a contextual identifier for managing shared states – the instance of the block within the course provides a common context for a shared state.

We were also able to create participants by passing user profile information to Wookie when instantiating Widgets. However, Moodle v1.9 does not have an events system we could use to remove participants, which meant that our test courses had large numbers of users, many of which were largely inactive. This was itself quite useful for testing purposes.

### 3.3 Widgets

For our prototype we converted the demonstration Google Wave Gadgets developed by Google\(^2\) and installed them in Wookie. This involved minor changes to the way the applications were packaged, to make them conform to the W3C Widgets specification. Otherwise no major changes were required. The Widgets chosen were a Sudoku game and “fridge magnet poetry”; some other ‘test’ widgets were converted, but only to check the API implementation was working correctly. The Google Widgets, while not perhaps profound examples, are nevertheless useful in demonstrating the features of live-updating collaborative Widget applications.

In addition to the Google Widgets, we also developed two of our own Widgets using the Wave Gadget API. One of these was a chat/wall Widget called Natter (updated from a previous version that used our own shared state APIs), a voting Widget called You Decide, and a word-making game called WaveWord. Developing these Widgets was relatively straightforward.

The following screenshot (Figure 3) shows all of these Widgets in action within a single Moodle course.

---

3.4 Embedding

As well as using the Widgets within the VLE, we also made it possible to take the Widgets, including all their contextual information, and embed them in other systems. Each Widget in the VLE also has a small “embed” icon underneath. Clicking this icon reveals a HTML code snippet that the user can place on another site (Figure 4).
Once the experimental system was in place, we invited participants to register and try out the experience. On June 12th, 2009, we announced the system in a blog post[8], on Twitter, in the Moodle developer forum, and on the Google Wave developer forum. Within a few hours we had acquired quite a few users. Initial issues with email-based registration meant giving out a default account for guests, and so we don’t have exact figures on the number of users. However, we estimate that around 200 users tested the system in the initial 7 days (12th-19th June); these comprised 124 named accounts, and at least 90 distinct IP addresses using the “moodler” public test user.

We were pleased to note that despite the system being open for public test, there were no performance, security or stability issues with the system, which ran with only a single restart, which was to update the Wookie code to a newer release. Users noted no major problems; one user reported an AJAX error, probably caused by a dropped connection, for one of the Widgets on one occasion.

Overall the performance results were very promising, as the prototype itself was running on a shared development server (running several other applications), with no special optimization in place (e.g. level two caching was not enabled for the database, and there was no special tweaking of our webservers for better Comet
performance). This gives us confidence that a large-scale demonstrator could be attempted to explore the performance boundaries of the system.

The Natter chat Widget also provided a means for us to capture comments from testers. The reaction was very positive; below is a small selection of comments received from testers.

- Very cool. I've got two browsers open and see my poetry changes live!!
- This is fun!
- Sat here in schiphol - if they get this wave tech right it will be incredible
- I like this effort
- Thanks for writing poetry with me
- It is very interesting I am gonna read this weekend some articles I took from Waveprotocol.org Just to understand the way it works
- I think wave is phenomenal! Can't wait ...
- This is a very nice demonstration of Google Wave and integrated into Moodle. Well done guys!
- Great stuff!
- This looks like it could be useful - need to get our VLE team on it
- This looks great much less static that the usual moodle stuff
- Can't wait to see more detailed documentation for how this is done - thanks for putting this demo up,
- Good work … Moodleusers luv ya baby!!!
- How cool!
- Very inspiring!
- I love it...can anybody help me to do this? Add wave in moodle?
- This looks good!
- Is there a whiteboard app?
- Do you know where I can download the source?
- Writing from the natter widget, embedded in my blog. Nice the way the highlights push it off the page. Now, let's see it appear over on the Moodle site...ah, here it is 😊

4 Analysis

What does it mean to have a “Moodle Wave”? While the technical work on integration did not bring about any particular surprises, the end result is a very different user experience from the conventional Moodle VLE.

4.1 Replacing VLE functionality

It was expected by the team that the use of Widgets, particularly those using the Wave Gadget API to provide live collaboration, would be a satisfactory replacement for the built-in tools of the VLE, such as chats, forums, and quizzes. This was certainly the case; and we believe that almost the entire toolset of a VLE can be readily replaced
with this type of Widget. Martin Dougiamas, the original developer of Moodle, made the following comment after the Google Wave video was released by Google[9]:

“Moodle (and similar systems) are (like it or not!) more about the *management* of learning at a higher level than this, a system of control and authentication that supports institutional policy and practice.

However, what I do think will happen is that many of the activities in Moodle will be replaced by services like these”[10]

This has significant implications for the VLE. If the majority of the features of the platform are readily outsourced to – possibly superior – external tools, then this requires a rethink of what defines a VLE.

For example, a version of Moodle that was developed with the assumption that it would be connected up with a Widget server would presumably not have large numbers of pre-developed PHP modules, but instead focus on areas outside the concern of individual Widgets, such as course and user management, integration and reporting to institutional systems such as the student record system, and perhaps integrating tools for structuring activities around Widgets, such as LAMS[11].

Would this more limited product profile make it easier for developers to create new, perhaps more innovative VLE-type products, at a more affordable price? Or would it instead make a case for bypassing standalone VLEs and instead integrate Widgets into more generic systems such as Sharepoint?

A counter-argument could be made that the existence of “cut and paste” embedding of widgets via JavaScript into the VLE is already possible, and has not had such an effect. However, there are considerable differences in the affordances of this approach versus the one we describe here. However a discussion of this point is beyond the scope of this paper.

4.2 New modes of interaction

Rather than just being a like-for-like replacement, the Widgets update in real-time without any page refresh, and this affects user behavior. Rather than clicking links to launch tools or to view content, the Widgets encourage more of a “monitoring” mode of operation, with users navigating to a course page, then leaving it open in the background, occasionally bringing it into focus to see if any new conversations were happening in the chat widget, or the voting results had changed.

This was visible in the activity reports for Moodle, which is conventionally used to see student activity. Rather than the usual histogram of page views seen in live courses, this showed a sparse distribution of visits to course pages. This may be an artifact of the type of informal testing we conducted, and so we identify this a hypothesis for testing in future pilots using the system.
4.3 Micro-tracking

Another consequence of Widgets is that far less tracking information is available at a micro-level, as interactions with Widgets are not made available to the VLE. This implies that a new model for tracking will need to be developed for such systems; this may be a useful opportunity to reconsider tracking in more sophisticated terms than page views and hits. For example, it may be useful to separate out measures of attention, using something like APML[12], and measures of user effort, using something like User Labor Markup Language (ULML[13]).

4.4 Embedding – the fuzzy boundary between institutional and personal systems

While the ability to embed Widgets from the VLE into other systems was not a significant aspect of this experiment, it was noted that several people had embedded Widgets from the example course into blog posts. One of the issues with embedding is that what for some people may be a “private” space could very easily spread into the public domain if participants embed Widgets from a course into public sites. The current architecture offers no practical means of enforcing restrictions on embedding, and so this would require policies and user education. A related issue is that it is hard to track where Widgets are being embedded; this could be addressed by placing tracking information into generated embed codes, which may also offer a solution to unauthorized public sharing. Striking an effective balance between enabling individuals to use PLEs, and supporting the rights of others will be one of the key challenges to be addressed.

5 Conclusions and Future Work

Our experiment demonstrates that it is feasible to embed live-updating collaborative applications within a VLE that can also be easily shared and embedded in personal technologies, and that this can be done using open standards and specifications. The Wookie technology supporting this experiment has been shown to be robust enough to support open testing, and is ready for a larger-scale demonstrator. However the challenges raised by this work are significant and will need to be addressed. In particular the tracking and monitoring functionality in VLEs – often posited as one of the key features of such systems – may be rendered partially or completely ineffective by Widgets. This should be examined in future studies, as if it is shown to be the case, existing tracking systems will need to be redesigned to cope.

We believe the approach taken in this work is going to be compelling for both students and teachers alike, and that this will challenge assumptions of what a “VLE” is and its relationship with other technologies. Eventually this could break down the dominant design of VLEs and open up institutions to more innovation in education technology.
References

Acknowledgments. The work on this publication has been sponsored by the TENCompetence Integrated Project that is funded by the European Commission's 6th Framework Programme, priority IST/Technology Enhanced Learning. Contract 027087


Towards Collaborative Portable Web Spaces

Stéphane Sire¹, Evgeny Bogdanov¹, Matthias Palmér and Denis Gillet¹

¹École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 
{stephane.sire, evgeny.bogdanov, denis.gillet}@epfl.ch
²Royal Institute of Technology (KTH), Stockholm, Sweden
matthias@nada.kth.se

Abstract. As two recent trends in Web development, Widgets and mashups, are converging, we claim that there is a growing need to define a Widget Space configuration language to allow building flexible personal learning environments that could be independent of a runtime environment. We describe four scenarios that could emerge from the integration of the do-it-yourself dimension of a Web portal or Web mashup and the social dimension of the Web. We also describe an extension to an existing configuration language that would support it and some propositions for an architecture and an implementation inline with recent developments and announcements such as the Google Wave Federation protocol.

Keywords: widget, mashup, collaboration, web, design scenario, PLE.

1 Introduction

End users of Widgets portals can compose their own personalized environment. Similarly, end users of mashups can benefit from components or applications created with one of the numerous mashup development environments such as Yahoo Pipes. Hence, it is not surprising that Widgets portals and mashups are amongst the most active areas of development today on the Web. For instance, as of July 2009, the mashups directory ProgrammableWeb (www.programmableweb.com) lists as much as 4000 mashups with 3 new ones registered everyday.

Similarly, the exponential development of social network Web sites shows that the social dimension, to bring people together, is one of the driving force that brings people to the Web.

These two trends are particularly compatible with two crucial outcomes of modern learning systems which are to let people construct their own learning environment, and to share their learning experiences together. For that reason, we claim that the future of Widget portals, mashup platforms, and social networks, is to develop a closer integration so that their users can create Widgets Spaces configurations that they can share at different coupling levels. This article explores the consequences of this idea.
2 Web Space Concept

2.1 Definition

Widget portals nearly always focus on providing a customizable personalized environment where various widgets are selected and organized by the user according to some principle. They provide both graphical layout and preferences for individual widgets.

A common approach, supported in iGoogle, Netvibes, Pageflakes etc., is to take widgets that are often accessed at the same time and put them into a separate tab. This could correspond to a course, a project, an interest, partaking in organizations etc.

In a similar way, mashup development environments allow to pick up software components which are connected together with different predefined settings. Some components are faceless, they are used to fetch, aggregate or filter data, while some others display input fields to allow user to enter data, and the output of the mashup can be visualized in a composite graphical view. This could support too a course, a project, an interest, etc.

In both contexts, we see the "tab", or the "mashup" as a more conceptual Web Space. This space is made of several components which are combined together, either graphically and/or through some kinds of event/data-flow wiring. Currently the usage of most Web Spaces is to gather information. However we see a great potential from a learning perspective in combining it with a social group to support collaborative activities as we will explain in the next sections.

In this article we see a Web space configuration as a three level information structure. The first level describes how to combine the components of a space (layout, graphical theme, data-flow, etc.). The second level describes the initial and default values of any preference or property of its components. Finally, the third level describes all the current values that were changed in the second level by the users since they started to interact with the Web space, and eventually some other user’s data which are stored within the space and not in external services.

2.2 Applications of the Web Space Concept

Once we recognize the Web space configuration as a basic template for instantiating a Widget composition or a mashup, and as a unit for storing user's data, it becomes natural to consider some mechanisms to keep it independent from the runtime environment through an adequate configuration mechanism. This supports three scenarios: portability, broadcasting and co-editing.
Fig 1. Scenarios enabled by decoupling the Web space configuration from the runtime platform.

The portability scenario allows a user of a Web space configuration, such as an iGoogle page, to move her Web space configuration to another platform, such as Netvibes, for example. We can also imagine that some adaptation to the Web space configuration applies to adapt it to runtime platforms with very different characteristics, such as converting it from a desktop to a mobile platform. The portability scenario also addresses the case when a user invites another user to duplicate one's own Web space, starting or not from a snapshot of one's personal data. This later usage could be called cloning.

The broadcasting and the co-editing scenarios are both scenarios where a connection is established over a long duration by the owner of a Web space that invites other users to share her Web space. In the broadcasting scenario, the owner invites other users to join one's own Web space and to keep them updated when she modifies some personal user's data in this Web space. In the co-editing scenario, the owner invites other users and all the changes to Web space done by any one are merged together so that there is only one Web space configuration.

The portability scenario can be combined with the broadcasting and co-editing scenarios in that different users sharing a same Web space can be accessing it from different runtime platforms. The co-editing scenario can be applied to a single user in the case where this user accesses her Web space from multiple platforms. Moreover, since widgets composition and mashups consist of several components and every component can have its own set of properties and user's data, sharing can happen at different granularity levels. For instance only some components may be shared, only some properties of some components, or some subsets of user's data. Sharing can also happen with different coupling levels depending on the frequency of synchronizations between the different instantiation of the Web space as we will develop in the next sections.

2.3 Current State-of-the-Art

To know to what extent our three scenarios based on Web spaces can be realized today, we have had a look at two widget platforms, namely iGoogle (www.google.com/ig) and Netvibes (www.netvibes.com), and at two mashups
development environments, namely Yahoo Pipes (pipes.yahoo.com) and Afrous mashup engine (www.afrous.com). We first realized that, to some extent, the Widget platforms already supports these scenarios at the individual widget level as explained in the table below.

<table>
<thead>
<tr>
<th>Table 1. Current support of the proposed scenarios at the Widget level.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Portability</strong></td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>iGoogle</td>
</tr>
<tr>
<td>Netvibes</td>
</tr>
</tbody>
</table>

The portability scenario is limited on iGoogle as it works only between users accessing the widget from iGoogle. In that case it appears as a "Send my settings for this gadget" checkbox when sharing it with someone. Netvibes seems also to provide an internal form of widget portability since there is a feature to archive a widget and to restore it at any time from/within Netvibes. Netvibes, as well as iGoogle, also allows you to embed certain widgets by generating a code snippet which can be cut-and-pasted into another environment (see the code below), however in that case the preferences and the configuration are hard coded into the code snippet. This type of portability is also limited to users familiar with HTML. The broadcasting scenario and the co-editing scenarios are supported in iGoogle, for certain widgets. The scenarios are materialized through a "View my content" and a "View and edit my content" options that can be checked when sharing a widget. However this is limited to sharing on iGoogle. A very limited form of co-editing is also available on Netvibes, as a user can access her widget from the desktop and the mobile version of Netvibes.

Widget level Portability scenario: code snippet to embed a Netvibes UWA widget.

```html
<scriptrc=""/UWA/load.js.php?env=BlogWidget2"></scriptrc>
<scriptr>
var BW = new UWA.BlogWidget(
{moduleUrl:'/multipleFeeds.php?provider=wiredblogs'});
BW.setPreferencesValues(
{"nbTitles":7,'showTweet':true, 'openOutside':false});
BW.setConfiguration(
{"title":"Wired Blogs", 'height':636});
</scriptr>
```

<table>
<thead>
<tr>
<th>Table 2. Current support of the proposed scenarios at the Web space configuration level.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Portability</strong></td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>iGoogle</td>
</tr>
<tr>
<td>Netvibes</td>
</tr>
<tr>
<td>Yahoo Pipes</td>
</tr>
<tr>
<td>Afrous</td>
</tr>
</tbody>
</table>
The portability scenario is supported at the Web space level on iGoogle and Afrous as they both allow to save and to import their Web space configuration into a file in different proprietary formats (respectively a gadgetTabML or a JSON file). Yahoo Pipes gives the possibility to clone a pipe from a mashup gallery to edit it.

The broadcasting scenario is supported on Netvibes as users can create one public profile page that their friends can see and which is synchronized with their changes. It is also supported on Yahoo Pipes and on Afrous. On Yahoo Pipes mashups can be embedded into "badges" which can be installed on other platforms through a code snippet that starts a Javascript runtime library (see code below). Similarly an Afrous mashup can be embedded through a Javascript runtime library, as the full mashup engine is a Javascript application. For Yahoo Pipes and Afrous, we consider the embedding as broadcasting as only a reference to the mashup is embedded, hence any modification made by the owner will be reflected in the embedded pipe. This applies to the data-flow composition and constant values parts of the configuration, and not to the user's data, as the concept of user's data (or preferences) is not explicit in these mashup platforms.

Code snippet to embed a Yahoo Pipe data mashup as a badge in a host page.

```html
<script src="http://pipes.yahoo.com/js/listbadge.js">
{"pipe_id":"AHkR4ai42xGlKUlZFUMqA",
"_btype":"list",
"pipe_params":{"results":null,"search":null}}
</script>
```

The current state-of-the-art shows that we have not yet fulfilled all the potentials of Web spaces, although our selection of 4 platforms shows that there is obviously a move into that direction. To go further, we will suggest in the next sections how we could proceed to define a common configuration language and an underlying architecture that would allow to fully support our 3 scenarios.

Currently the most advance move into that direction that we are aware of is the gadgetTabML format that Google is using to export and import personal pages on iGoogle, and that looks like an undocumented internal feature. The OPML file format is more widespread but it only handles feed widgets in tabs, all other widgets and their preferences are lost. The mainstream widget standardization committees, such as W3C, Open Ajax Alliance, and Open Mobile Terminal Platform (OMTP) do not seem to have plan in that direction, and neither do mashup development environment providers.
3 Web Space Configuration Language

3.1 Configuration Language Elements

A Web space configuration language should at least define the following elements:

- **List of widget or mashup components with default or user settings**: the list refers to components which are published openly and which can be easily referenced via URIs. Depending on the component format, the settings may be expressed slightly differently. Unless the component settings are exposed as properties, it is not stored in the Web space. Components may therefore have separate dependencies to cloud services or real world information for user data storage, but in that case they should rely on other services to share these data.

- **Layout of graphical components**: since not all Web space container support the same layouts, the layout should be treated as hints.

- **Event and/or data-flow wiring of components**: mashups are created by connecting their components, this is also becoming part of a widgets composition as mechanisms such as inter-widgets communication and drag and drop become available [5]. These connections or communication channels are described here.

- **Participants list**: it reflects the social context in which learning occurs. In our vision, one owner of a Web PLE built on top of Web spaces will invite other learners to share an initial configuration associated with some learning goals. That initial configuration template will be created by a teacher for instance, and corresponds to a scenario (e.g. "get to know each other"). The list of participants defines the scope for broadcasting or co-editing when the space is shared.

- **Sharing list**: in order to broadcast or co-edit a space, the sharing list defines precisely the basic units of sharing such as individual properties of a component, a full component, or other properties such as the position of each components in a given graphical layout for instance. We suggest as a sharing policy that all the properties not explicitly shared can be changed by each user independently of the others.

- **Refresh rate list**: programming patterns such as Reverse Ajax allow to develop notification mechanisms in such a way that changes to shared properties can be notified synchronously. The alternative is to refresh them only on page reload. The refresh rate list indicates the preferred mode of update for each shared properties.

A configuration can be seen as a container that points to external configuration elements, such as the widget configurations. This is a template that supports the creation of a new Web space from external resources. However, the configuration should also store all the shared settings which have been updated by the users in order to distribute them to the group. In this regards we currently describe these settings as properties (i.e. key/value pairs), which is compatible with the practices in most widget and mashup platforms. These properties have many different usage in actual
environments. Some properties serve as user interface preferences, some others as widget state variables, and finally some others as application data storage. Thus the interest of broadcasting or co-editing a space may depend greatly on the way its components define and use properties.

Finally it appears that we have three orthogonal dimensions in the configuration: participant lists, sharing rights over a property, and refresh rate. The combination of these three dimensions can be used to achieve different styles of user interface coupling [3].

### 3.2 Configuration Language Syntax

As an exercise we have started to imagine how we could extend the gadgetTabML configuration files so that they can configure different broadcasting and co-editing scenarios. Our proposition is to use a new XML element and 3 new XML attributes (see table below) which can be set on the different elements of a host configuration language to decide who can see them, how they are shared and at which refresh rate.

<table>
<thead>
<tr>
<th>Attribute or Element</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;Participants&gt;</code></td>
<td>user identifiers (e.g. email)</td>
<td>a list of persons sharing a Web space configuration</td>
</tr>
<tr>
<td>@participants</td>
<td>user identifiers</td>
<td>a list of persons viewing a particular component</td>
</tr>
<tr>
<td>@sharing</td>
<td>BROADCASTING (a) EDIT (b) NO (c)</td>
<td>type of sharing, (a) means only the owner can change the value, (b) means everybody can change it, (c) it's a private property</td>
</tr>
<tr>
<td>@refresh</td>
<td>SYNC (i) ASYNC (ii)</td>
<td>delay before updating the shared properties, (i) means synchronously, (ii) means on page reload</td>
</tr>
</tbody>
</table>

The following example shows a space configuration with 4 widgets that Alice has shared with Bob, Charlie and Dave which are declared in the `<Participants>` element. According to that configuration Alice broadcasts the xmlUrl property of the first RSS widget (i.e. she imposes the feed). This is declared with the sharing attribute set to "BROADCAST" on the `<ModulePrefs>` declaration of the property. The updates that Alice will make to the feed URL fill be broadcasted asynchronously (i.e. on page reload) because of the refresh attribute set to "ASYNC".

Alice has chosen to display the GoComics widget only in the view of Bob. This is specified with a participants attribute on the `<Module>` that declares the GoComics widget. Alice has also chosen to impose the value of the "mycomics" `<UserPref>`
property to Bob (attribute sharing set to "BROADCAST") so that she can show him her own choice of comics.

The second RSS widget is visible by everyone and everyone can define his own settings because there is no sharing attribute defined. Finally, all the properties of the Weather widget user preferences, except the "temperatureUnit" (attribute sharing set to "NO"), are shared and can be edited by everyone because its sharing attribute is set to "EDIT", and updates are synchronous.

Extended GadgetTabML file for usage as a Web space configuration (some namespace declarations and some URLs have been removed for simplification.

```xml
<GadgetTabML version="1.0" xmlns=".../GadgetTabML/2008">
  <Tab title="Accueil" skinUrl="skins/sampler.xml">
    <Layout iGoogle:spec="THREE_COL_LAYOUT_1" />
    <Participants owner="alice">alice@kth.se bob@epfl.ch charlie@epfl.ch dave@kth.se</Participants>
    <Section>
      <Module type="RSS">
        <UserPref name="numItems" value="3"/>
        <ModulePrefs xmlUrl="http://..." sharing="BROADCAST" refresh="ASYNC"/>
      </Module>
      <Module type="REMOTE" participants="alice@kth.se bob@epfl.ch">
        <ModulePrefs url="http://gocomics/.../gc.xml"/>
        <UserPref name="defaultIdx" value="1" />
        <UserPref name="selectedTab" value="comments" />
        <UserPref name="mycomics" value="9|32" sharing="BROADCAST"/>
        <UserPref name="favoritesIdx" value="0" />
      </Module>
      <Module type="RSS">
        <ModulePrefs xmlUrl="http://..."/>
      </Module>
    </Section>
  </Tab>
</GadgetTabML>
```
<Module type="WEATHER" sharing="EDIT"
    refresh="SYNC">
    <UserPref name="temperatureUnit" value="F"
        sharing="NO"/>
    <Location name="New Orleans" country="US"
        language="en"/>
    <Location name="San Diego" country="US"
        language="en"/>
</Module>
</Section>

This is a simple proposition to map the participants, sharing and refresh rate dimensions onto a host XML-based Web space configuration language. Of course there are several directions in which the proposition can be refined. For instance, we could imagine a syntax to declare if users can add new widgets/components, if they are shared, and if they can be removed from the space. Similarly we could define a syntax to define some other configuration elements that could be shared such as the layout. Finally, other extensions are necessary to allow finer grain access control models with sub-groups of users with broadcasting capabilities, and not just use the owner/others dichotomy.

4 Architecture for Collaborative Portable Web Space Configurations

Usually widget and mashup platforms are divided between a server component, that we call the engine, and a client-side component, that we call the container. The engine is mainly responsible for maintaining the Web spaces configurations, usually in an internal database. The widget or mashup components source code may also be installed on the engine server, or they may be available from third party servers. The container is responsible for visualizing the Web space inside a browser, or maybe within another Web space in case portability is offered outside of the platform with compatibility Javascript libraries as we have described in the state-of-the art.

We propose to add a new element to this architecture, namely, a Configuration Server. This component is required to make a Web space portable, or to start sharing it. To simply clone a Web space, an engine just needs to copy the current state of its
configuration to the configuration server. To start sharing it, the engine needs to communicate with this configuration server to maintain the configuration up to date as defined by the participants, sharing and refresh settings of the configuration.

![Diagram](image)

**Fig 2.** Two visions of the Portable Web Spaces architecture.

This architecture admits two variants. In the first variant, a single, centralized configuration server has been setup to allow two users to share a configuration from different widget platforms. In the second variant, the configuration data is hosted on different federated configuration servers. The second variant is more flexible in that it allows portability or sharing between platforms that may be connected with different configuration servers, or to have users with different configuration servers in case they need to have an account on it before using it.

5 **Implementation Issues**

In this section we simply outline some ideas concerning the possible implementation of a Web space configuration server.

The Web space configuration server only needs to manage configuration data, thus it can be implemented with any kind of database. To support sharing, it must also propagate data updates. One solution for synchronous updates is an extension of the XMPP protocol: Google Wave Federation Protocol [1]. In that scenario, the configuration server would implement/utilize this protocol and create a new wave for every Web space configuration. Then, every property of a configuration with a synchronous refresh rate could be saved as a new wavelet into the Web space wave. The participants list of the space would be copied to the participants list of the wave.
If the refresh rate is asynchronous the property would simply be saved into the configuration server database.

The engine would communicate with the configuration server with a protocol to be defined, maybe using the Google Wave client-server protocol [2], not only for synchronous properties but also for asynchronous ones. In all the cases, the implementation requires to adapt all the existing platforms so that they can use an external configuration server instead of their internal database.

The container would communicate with the engine using COMET or Reverse Ajax methods or higher level APIs such as the Realtime Gadgets API [4] or HTML 5 Server-Sent Events[6].

Then, two options are possible to manage the sharing policy of the different properties. The lightest solution, that would not require a modification of the engine, is to manage the sharing mode at the configuration level. However, this would not allow to support the broadcasting mode, as everyone could actually change the property values locally in her browser and not only the owner. The second option is to manage it at the engine level, so that it prevents users from modifying a shared property if it is broadcasted and they are not the owner. In both cases it would also be necessary to modify the container part of the platform, since the user interface should provide specific affordances and feedbacks to the users so that they know what is shared and how. To some extent the widget and graphical components developers too should also take this into account. Of course, the widget or component API must also be modified to take into account property changes triggered by remote users, for instance through a callback mechanism.

Regarding the social dimension, which is introduced through the participants list, a potentially interesting solution could be to delegate the management of groups of people to an external service such as a kind of OpenId but for groups, let's call it GroupId. It should go beyond the definition of a simple list of friends, as defined in an OpenSocial container, to support the management of several groups and subgroups a user belongs too. A group managed by a GroupId server could be declared as a participants list in a configuration through a GroupId URL. This way it should be easier to support many collaboration models, for instance to define open/public participation lists, without changing the Web space configuration language.

6 Wrap up example

The following example shows a concrete application of portable collaborative Web spaces. A teacher has a class with five students. The task of the students is to develop a computer program collaboratively. The program should consist of four main blocks, each solving a particular problem, and a block that combines the code to solve the task. He decides to use the following settings for students: a co-edited widget, that contains the final code, and a widget for every user, where the particular task is specified and a student can do some unit testing on his own code. Thus, the Web space contains these six widgets.
The co-edited widget is available to all users and access to the rest five widgets is limited to one user. Even though the teacher decides to create his Web space in iGoogle, some of the students prefer to use Netvibes. Since the Web space configuration is available to all Web space servers, students can use any environment they want. The changes to the co-edited widget are propagated synchronously to all the users and the teacher. Later teacher discusses the successful class with his fellow teacher from another university. The latter likes the idea, and in order to try it out he receives the clone of the Web space configuration to try it in his Pageflakes portal.

7 Conclusion

This article introduces the notion of a Web space as a basic unit for storing settings and personal data of a Widget composition or a mashup, which can then be independently managed by a configuration server. We have shown that this decoupling between the configuration and the execution of a Widget composition or a mashup enables portability, broadcasting and co-editing scenarios. As we have illustrated, the current state of advancement of the Web is not far from realizing this vision. Thus, we invite every interested party in agreeing on the standards and formats that would allow to construct such a configuration server, maybe on top or as extension of existing technologies such as the Google Wave Federation Architecture or some OpenID extension.

Acknowledgement

This work has been co-funded by the European Union under the Information and Communication Technologies (ICT) theme of the 7th Framework Programme.

References

4. Google: Realtime Gadgets API (2009), online code.google.com/apis/talk/gadgets_realtime.html

Infusing reflective practice in eLearning courses – Can widgets help?

Dominique Verpoorten, Wim Westera and Marcus Specht

CELSTEC, Open University of the Netherlands, Valkenburger Weg 177, 6411 AT Heerlen, The Netherlands
{dve, wim, spe}@ou.nl

Abstract
This paper explores the potential of widgets for reflection, viz. instructional widgets that would be designed to prompt and support students' reflection in eLearning courses. The investigation builds on a literature review, an inventory of 35 reflection techniques and a survey amongst eLearning course owners. The outcomes demonstrate that a large part of the reflection techniques are acknowledged by teachers as being of relevance for their courses. Yet, practical application in distance education is quite rare. Results of the survey are used to ascertain possible contribution of software widgets to the implementation of some of these reflection techniques.

Keywords: eLearning, Widgets, reflection, Open Content, meta-cognition

1 Introduction

This article is positioned at the cross-section of a new Internet technology and a new pedagogical trend by combining a new breed of software applications called widgets with the call for more reflection in learning.

As for the first trend, the term "widget" refers to a miniature Web application performing a single task and displaying a very clear and appropriate graphical style [1]. It provides a single interaction point for the visualization and direct manipulation of a given kind of data [2]. Typical examples would be widgets that show today’s weather forecast, upcoming birthdays or information stocks, designed for the desktop, the Web or the mobile. Personal learning environments are already taking advantage of widgets. More broadly, the technical approach based on widgets has become available to eLearning in general whilst it is not yet clear how they can best be used within a formal context of instruction. This paper opens a line of inquiry about "widgets for reflection", namely widgets designed to prompt and support clear, small and single reflection-related tasks occurring prior, during or after a learning sequence.

With regard to the second trend, reflection is an active process of witnessing one’s own learning experience and evaluating it on different aspects. Reflective practice (and akin notions like "learning to learn" and "meta-cognitive development") is a highly influential factor of learning [3, 4, 5, 6], enhancing both the domain-specific knowledge and the knowledge about the self-as-a-learner. However, current
instruction shows a shortage of training for this generic skill [5, 7, 8]. Quantity, quality and persistence of reflection might substantially go up if the offer of tools for reflection could be enriched. In this respect, the option of "widgets for reflection" would nicely align with the call for more reflection in learning. This paper presents a survey amongst teachers on techniques for reflection. It uses the outcomes to identify which of these techniques are feasible candidates to an implementation as software widgets. It discusses the potential of these specialized widgets in transforming a learning environment so that it can become supportive for meta-learning training.

2 The survey: Reflective Practice in eLearning courses

The survey was carried out amongst 22 owners of an eLearning course who partook in the Open Educational Resources project (OpenER) of the Open University of the Netherlands [9]. The OpenER-project makes available a variety of higher education eLearning content free of charge. Like similar initiatives over the world (MIT OpenCourseWare, MERLOT, OPENLEARN, etc.), it targets an expansion of the higher education learning opportunities. The choice of OpenER courses for the survey has three reasons. First, the Open University of the Netherlands has defined a program aimed at enhancing its offer of open educational resources. Second, the course owners are experienced developers of eLearning content. Third, the research was conducted in the context of the i-Coper project which is dedicated to open educational resources. Course owners were asked to fill in an electronic questionnaire in regard to opportunities for reflection displayed in their courses. The survey is presented below.

2.1 Objectives of the survey

The objective of the survey was threefold:
- to investigate the state of affairs of reflection amplifiers in OpenER courses. Reflection amplifiers are defined as structured opportunities for students to examine and evaluate aspects of their learning experience;
- to give insight about the relevance of reflection in the eyes of instructors;
- to ground the discussion about harnessing widget technology to instructional objectives into data coming from practitioners.

2.2 Method and presentation

In March 2009, 22 owners of an Open Educational Resource course of the Open University of the Netherlands answered to an online questionnaire which presented the description of 35 existing techniques meant to amplify reflection, as reviewed and categorized by Verpoorten, Westera & Specht [10]. For each reflection amplifier, respondents were asked to tick one of the following options: I do not understand this technique / This technique is not relevant for my course / This technique would be relevant for my course but is not implemented / This technique is implemented in my course. This type of investigation was chosen in order to find out the extent to which
reflective approaches to learning are implemented in existing courses, and which concrete techniques are considered as relevant by faculty. The research was exclusively based on the answers given by participants. No reality check was done in the courses. The small size of the sample, the subjective dimension of the answers and the fact that the survey was the entry point to a larger investigation concerned with meta-learning practice in formal education account for the presentation of results in plain numbers and not in more sophisticated measures and correlations.

2.3 Results

Overall, 13 course owners (cfr. Fig. 1) out of 22 completed the questionnaire. In view of the 35 reflection amplifiers that were presented, this means that the study collected 455 (13 x 35) practitioners' qualifications over reflection amplifiers. Below, the aggregated outcomes are summarised.

**Understanding of the reflection amplifiers.** The reflection amplifiers seem to be well understood. Only 23 out of 455 answers "I do not understand this technique" were collected. Amplifiers the least understood, that is amplifiers for which the option "I do not understand this technique" has been the most often ticked, are Formative assessment (4/13), Structure for regulative support (3/13), On-demand assessment (3/13), Confidence-Based Learning (3/13).

**Relevance of specific reflection amplifiers.** Respondents, 73 times out of 455, claim that a specific amplifier would be relevant for their course but is not implemented. Amplifiers with the most potential are: Help seeking behaviour guide (4/13), Graphical presentation of contents (4/13), Students set the test (4/13), Indicators of understanding (4/13).

**Overall relevance of reflection amplifiers.** When grouping the answer categories "This technique would be relevant for my course but is not implemented" and "This technique is implemented in my course" versus "This technique is not relevant for my course", it gives 157 claims of relevance versus 275 claims of non-relevance. (The 23 "I do not understand this technique" are not taken into account). So, a large part of the reflection techniques are recognised as being of relevance for the eLearning courses.

**Existing practice.** According to respondents, 82 out of 455 reflection amplifiers are implemented in the courses. Highest occurrences: Making pedagogical rationale transparent (9), Metacognitive modelling (8), Self-explanations (6), Practice of evocation (4), Justify your choice (4), Graphical presentation of contents (4), Room for choice (4). Used amplifiers are unevenly spread in the courses (cfr. Fig. 1).

We analysed data to find patterns of aggregation of reflection amplifiers but no significant one could be identified, not even at the level of one-one combinations.
Focus on the exploitation of tracked data. Several reflection amplifiers are based on the mirroring of personal tracked data. The study reveals that 7 out of 13 course owners don’t know whether the eLearning platform on which they developed the course provides any tracking facility. Overall, 3 respondents state that they use tracked data as a teacher. Only 1 respondent says that the tracked data is used by the students. When asked whether they would give their students access to their learning traces as a reflection amplifier, 4 teachers out of 13 answer positively (cfr. Fig. 2).

3 Designing Widgets as reflection enablers

The questionnaire survey was carried out to investigate the practitioners’ views about reflection in eLearning courses. From this empirical output, four observations are derived regarding possible contribution of widget technology to the facilitation of a
reflective approach to learning. These observations should be seen as working hypotheses that ought to be further refined and tested.

3.1 Widgets for reflection can capitalize on interest for reflective practice

Supported by literature about the relevance of reflection skills, widgets for reflection may also find supporters amongst teachers. According to the present study, course owners express a fairly high rate of relevance regarding techniques for fostering reflection (157 out of 432). This may even be an underestimation because of the specific context of open educational resources. Some of the respondents used the open comments section in the questionnaire to express practical disclaimers for not using reflection amplifiers: three respondents stressed that they had limited time and resources to devote to the design of the course. They also state that the request of the Open Educational Resources project was just striving for digital content in order to have a few courses available as soon as possible. Despite these adverse conditions, 36% of the reflection amplifiers submitted to the whole group are ticked as relevant.

3.2 Feasible candidates to “widgetisation” can be found

Teachers gave their opinion about 35 reflective techniques. Most of these techniques are too complex to be used as widgets, according to the definition given in the introduction. Amongst others, this probably holds for "On-demand assessment", "Portfolios", "Students set the test", "Help seeking behaviour guide", etc. However, "widgetisation" seems possible for the following techniques.
- Growing progress visualization tool: the widget would offer visual displays (e.g. progress sliders, understanding meters) enabling learners to determine their progress (actions and mastery) towards the learning goals. 3 respondents out of 13 consider this feature as relevant for their course;
- Comparison with yardstick: the widget would specialise in comparing certain aspects of the learning process (time spent, exercises completed, estimation of knowledge, own performance, etc.) with some yardstick (teacher, peer, expert, classroom average, oneself in similar circumstances, compliance ratio, etc.). 7 respondents out of 13 consider this feature as relevant for their course;
- Indicators of understanding: the widget would prompt learners to qualify their understanding of the course with simple indicators like "lost/foggy/got it" or some similar labels. 7 respondents out of 13 consider this feature as relevant for their course;
- Judgement of learning: the widget would allow students to report the progress they believe they made in the learning domain or objectives as a consequence of doing the course. 7 respondents out of 13 consider this feature relevant for their course;

It is not excluded that changes in the technology may make a different definition appropriate (such an evolution of definition can be observed, for instance, with learning objects) or that advanced widget-like techniques could address these complex techniques or complement them (it could, for example, be the case with smart indicators [11, 12]).
- Self-efficacy judgments: the widget would engage students in self-assessments of their perceived level of knowledge or ability for a task. 7 respondents out of 13 consider this feature as relevant for their course;
- Mirroring of personal tracked data: the widget would allow a visualisation by the learners of different interactions they had with the course. 3 respondents out of 13 consider this feature as relevant for their course.

Once developed and embedded in the courses, such specialized widgets would represent self-contained learning activities. Each of them could be formalised as a triplet [13, p. 3] of:
(a) at least one tool. Example: "I use the widget "Understanding indicators";
(b) an action. Example: "With the widget, I rate my understanding of this content";
(c) an outcome. Example: "Thanks to this widget, and through the clear, small and single action it allows, I train my meta-learning skill for self-assessment".

3.3 A pick-and-mix, progressive and cheap approach to reflective practice

No single outstanding reflection amplifier emerges from faculty's answers and no preferred combination either. It means that teachers can just pick out one or several techniques as relevant (cfr. Fig. 1). The modular approach conveyed by widget technology, and more broadly by Web 2.0, seems quite suitable to cater for these variations. An individual teacher could select and aggregate (or not) widgets for reflection according to the courses, the student's needs or the level of reflection to be pursued. In such a pick-and-mix approach, inclusion of opportunities for reflection, within the framework of regular instruction [8, 14], might be tailored and progressive. Moreover, such a widget-driven evolution of already existing courses is more conceivable and cheaper than introducing new courses. Also, when applying these tiny and not much disruptive appliances, users (teachers, learners) are shielded against the need of dropping the existing VLE and having to get acquainted with a completely new system. Furthermore, long-lasting results cannot be expected from one or even a few reflection exercises. Many of them must take place during the academic year, over several years and throughout different courses. The agile nature of widgets for reflection might ease this deployment and concur to the acquisition of reflective habits. The survey also suggests that research on widgets has a value on its own and does not need to be tightly coupled with research into Personal Learning Environments. eLearning courses can be relevant containers for widgets.

3.4 Widgets and tracking and tracing data can be mutually supportive

The survey delivers ambiguous answers regarding contemplation of personal tracked data as a lever for student's reflection. On the one hand, to have students pondering upon their interaction footprints is granted some potential by practitioners (cf. Fig. 2). On the other hand, 7/13 course owners don’t know whether their eLearning platform
provides any tracking facility. They do not use the traces themselves and do not know whether students do. Several studies indicate that teachers [15, 16], students [17] and learners [18] can reap meta-learning benefits from the observation of their traces. Making this data available through specialised tracking and tracing widgets is likely to boost the extent of this practice. It implies to establish links between tracking and tracing facilities and specialised tracking and tracing widgets. From an application point of view, such widgets would remain single objects but their semantics, visual appearance, dependencies and overall development can become very complex and demanding, as already observed in an early article on the topic [19, p.3].

4 Conclusion

Looking at reflection as a desirable educational goal induces the quest for instruments that are likely to foster it. This article considered the possibility of harnessing the agility of widgets to the training of thinking skills, within the framework of subject matter instruction. Based on literature, an inventory of 35 reflection techniques and a small-scale survey amongst teachers, it was argued that the development of widgets for reflection is a promising means to the infusion of certain types of reflective practice in eLearning courses. Due to its specific features – agility, interoperability, self-contained activities, aggregation power – widgets technology seems especially appropriate:
- to increase opportunities for instant and focused reflection within a particular learning task;
- to support an extended training of auto-cognitive skills (awareness during study, self-assessment, presence-to-learning) by embedding widgets for reflection within a variety of courses and systems;
- to provide teachers with ready-to-use reflective tools likely to be seamlessly activated according to the configuration they find the most pedagogically relevant;
- to facilitate cognitive regulation of personal learning by providing coordinated access to a variety of personal tracked data.

These promises should now be transformed into proper examples and associated hypotheses which, in turn, should be tested through additional research. In this respect, the authors are currently creating an eLearning course prototype enriched with concrete examples of widgets for reflection. This course prototype is meant to provide a convenient context for research on conditions of use, impact and possible drawbacks and benefits of these pedagogically and technologically innovative learning tools. At this initial stage, widgets for reflection will be developed for the Moodle platform as "Moodle modules". Technical feasibility, possibilities of truly smooth integration, acceptability and familiarity conditions will be studied. Through experimental setups, it is planned to explore effects of various displays, groupings, sequencing, and coordination of reflective tools on different dimensions of the instructional design and learning processes. If the experiments carried out in this particular environment deliver evidence of instructional benefits that buttress the interest expressed by teachers, issues related to the interoperability of the useful widgets in various learning environments will be addressed.
References

The MUPPLE Competence Continuum

Joanna Wild¹, Fridolin Wild³, Marco Kalz², Marcus Specht⁴, Margit Hofer¹

¹ Centre for Social Innovation (ZSI), Austria, {hofer,wild}@zsi.at
² Open University of the Netherlands, The Netherlands, {marco.kalz,marcus.specht}@ou.nl
³ The Open University, Milton Keynes, United Kingdom, f.wild@open.ac.uk

Abstract. The idea of Personal Learning Environments (PLEs) seems to polarise the educational sphere into supporters and opponents. Both groups relate their enthusiasm or criticism to underlying competences motivated by or needed for building up, running, and maintaining a PLE. Within the following article, results of a qualitative study with multiple cases will be presented to shed light onto which competence and which of its building blocks are involved in running a (mash-up) PLE. Data about the involved skills, abilities, habits, attitudes and knowledge will be presented in a raster of the five dimensions 'plan', 'reflect', 'monitor', 'act', and 'interact' against the three stages 'start', 'trigger', and 'outcome'. The findings indicate that there is a continuum ranging from the ones needed right ahead to the ones ultimately sought.

Keywords: PLE, mash-ups, competence.

1 Introduction

In the beginning personal learning environments (PLE) were strongly motivated by their opposition to learning management systems, while today they form a movement on their own. PLEs provide a perspective on learning environments that focuses on the individual (not the institution): they envision an empowered learner aiming for self-direction for whom tightly- and loosely-coupled tools facilitate the process of defining outcomes, planning their achievement, conducting knowledge construction, and regulating plus assessing (van Harmelen, 2008) – either collaboratively or independently. Mash-ups relate to the ‘frankensteining’ of software artefacts and data. They serve end-users to glue together public web services in individual applications (Wild et al., 2008). Their combination with PLEs merely stresses the fact that the digital parts of a PLE are today typically web-based, distributed across a variety of web servers, and networked.
While there is currently an intense discussion about the concept of personal learning environments and its technological and organisational foundations, it remains an open question whether a particular set of skills or competencies is required for facilitators and learners to use PLEs in their education or educational design activity. This question is strongly related to the ‘media literacy’ and ‘digital literacy’ discussion in educational technology. Thoman & Jolls (2005) define media literacy as a “21st century approach to education. It provides a framework to access, analyse, evaluate, and create messages in a variety of forms – from print to video to the Internet. Media literacy builds an understanding of the role of media in society as well as essential skills of inquiry and self-expression necessary for citizens of a democracy”.

In their declaration on lifelong learning from 2006 the European Commission mentions eight key competences. One of these competences is defined as a “digital competence [that] involves the confident and critical use of information society technology (IST) and thus basic skills in information and communication technology (ICT)” (European Parliament and the Council of Europe, 2006). Furthermore, a digitally literate person is “equipped with the skills to benefit from and participate in the Information Society. This includes both the ability to use new ICT tools and the media literacy skills to handle the flood of images, text and audiovisual content that constantly pour across the global networks” (EC, 2007). Learners with a high level of digital literacy deploy information and communication technology efficiently depending on situation and aim, use them to generate information and knowledge in their profession, and transform knowledge and practice through innovation and creativity with the help of these technologies (DTI, 2007).

DTI (2007) argues that in opposite to digital literacy that is usually defined in terms of technologies, media literacy is defined in terms of contents (e.g. images, text, audiovisual) of which the skills in question are to allow mastery, see Figure 1. Also, instrumental, informational, and strategic skills – i.e. the ability to operate hardware and software, the ability to search for relevant information using hardware and software, and the ability to use information for own purpose and position - are all equally important for digital literacy whereas media literacy is primarily concerned with strategic skills which include the ability to evaluate and create messages (DTI, 2007).

A model of so called eSkills is proposed by Stucky et al. (2003). Their IT Competence Maturity Model categorises eSkills according to five degrees: (1) IT awareness (basic knowledge), (2) IT literacy (knowledge to operate a PC), (3) expert user (special knowledge or expertise with application software, helping other users), (4) professional entry level (professional knowledge) and (5) professional level IT skills (advanced professional knowledge). Different than media and
digital literacy, eSkills primarily focus on instrumental aspects and provide a model for developing proficiency in applied settings (see Figure 1 below).

![Figure 1. State of the art: overview.](image)

With the rise of social software in the last years and technologies for a programmable web like Ajax, a silent revolution has taken place that promotes a new era of end-user friendliness and usability that should enable an average user to access, manage, re-mix, create, and communicate information and knowledge to various individuals and networks more efficiently. At the same time public exposure and the ever-growing collection of own digital traces became a source of valuable information for constant improvement of the user’s own performance.

It is still, however, an open question, which skills are needed to apply these technologies in an educational context. The here presented empirical study aims to shed light on the competences (skills, abilities, attitudes, habits, knowledge) required and motivated by the PLE and – more specifically – Mash-Up PLE approaches. We believe that our findings provide some valuable input on how literacies and skills are developed with the help of these environments. Furthermore, the findings allow to connect the MUPPLE and PLE debate to the ongoing media- and digital literacy debates.
2 Methodology

A qualitative research approach, particularly case study research, was applied as the key methodology in this study. Qualitative research methods are designed to help researchers understand people and the social and cultural contexts within which they live. Kaplan and Maxwell (2005) argue that the goal of understanding a phenomenon from the point of view of the participants and its particular social and institutional context is largely lost when textual data are quantified.

The data for this investigation was collected in a series of interviews with representative people from the target groups facilitators, learning designers, learners, and researchers. All interviews were open-ended interviews, but they were roughly structured by a set of guiding questions. The interviews focused on identifying competences involved in PLE use of both learners and facilitators. This article only presents the findings about the learners. The length of the interviews varied between 40 minutes to one hour. Three sets helped to triangulate the subject under study in order to make it less subjective and in order to incorporate multiple stakeholder views.

The first interview partners were a group of eight European instructors attending a training workshop on how to use social software tools for learning delivered by the REVIVE project in March 2009 in Budapest. Six of the interviewees represent higher education institutions; two design e-learning courses for a chamber of commerce. All interviewees have previous experience in using a virtual learning environment for teaching, whereas only four declared to have some experience with social software tools such rss-feeds, blogs, google docs and social-bookmarking tools.

The first interview was followed-up by a (virtual, i.e. skype) focus group session comprising three experts on educational technology, followed by an expert interview with an educational psychologist. The picture has been rounded up by a (virtual, i.e. flashmeeting) interview of a group of three media pedagogy students. All interviews were recorded and a transcript was prepared which was used as data basis for the analysis.

Subsequently, analytical memos were elaborated sessions from the gathered data by the investigators, the content thereby grounded in and inductively distilled from the data. Coding was performed to group identical and strongly related concepts. Memos and codes were finally (re-)arranged in a large display to identify and elaborate the proposed conceptual structure of five dimensions along three stages.
**Fig. 2. The PLE competence continuum.**

<table>
<thead>
<tr>
<th>minimal condition</th>
<th>necessary trigger</th>
<th>intended outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>plan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>inquisitiveness</td>
<td>ability to explicite intentions and objectives</td>
<td>ability to design your own portfolio</td>
</tr>
<tr>
<td>offer-oriented selection</td>
<td></td>
<td>ability to match formal and personal requirements</td>
</tr>
<tr>
<td>adopting 3rd party planning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reflect</td>
<td></td>
<td></td>
</tr>
<tr>
<td>willingness to change own attitudes</td>
<td>collecting (digital) traces</td>
<td>ability to identify strengths and weaknesses</td>
</tr>
<tr>
<td>monitor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>adopting external evaluation criteria</td>
<td></td>
<td>control &amp; direct facilitator(s)</td>
</tr>
<tr>
<td>systemic ‘more/knowledgable-o/other’ evaluation</td>
<td></td>
<td>self discipline</td>
</tr>
<tr>
<td>act</td>
<td></td>
<td></td>
</tr>
<tr>
<td>willingness to try s.th. new</td>
<td>capability to deconstruct your PLE</td>
<td>productive and transformative ICT skills</td>
</tr>
<tr>
<td>instrumental skills: basic operational ICT skills</td>
<td>ability to screen available tools</td>
<td>ability to assess quality &amp; reliability of sources of information</td>
</tr>
<tr>
<td>information gathering skills: content-centred</td>
<td>capability to repurpose change tool context</td>
<td>process &amp; content-centred</td>
</tr>
<tr>
<td>interact</td>
<td></td>
<td></td>
</tr>
<tr>
<td>social interest</td>
<td></td>
<td></td>
</tr>
<tr>
<td>basic social skills</td>
<td>willingness to open artefacts up to others</td>
<td>networking competence</td>
</tr>
<tr>
<td>(foreign) language skills</td>
<td>willingness to give</td>
<td>ability to network for feedback</td>
</tr>
<tr>
<td></td>
<td>willingness to not only read, but publish</td>
<td>decision competence</td>
</tr>
<tr>
<td></td>
<td>extended social skills: knowing how to handle mediated</td>
<td>when to work collaboratively</td>
</tr>
<tr>
<td></td>
<td>communication &amp; criticism</td>
<td>negotiation skills</td>
</tr>
</tbody>
</table>

### 3 Results

Within the conducted interviews, five dimensions with three stages each could be identified. These dimensions encompass: plan, reflect,
monitor, act, and interact. The stages we could distinguish so far partition the underlying competences (skills, abilities, attitudes, habits, knowledge) into those that serve as a minimal condition, necessary triggers, and intended outcomes.

With 'minimal condition' we refer to those skills, abilities, attitudes, habits, and knowledge that need to be present or are typically present when beginning to consciously build a PLE, whereas 'necessary triggers' denote those that are developed along the way towards the 'intended outcomes' and on which the intended outcomes rely. They can be surrogate or just incomplete in nature.

In brief, planning competence refers to those skills, abilities, habits, attitudes, and knowledge that fix how goals, schedules, and paths are set. Reflection is creative sense making of the past and enables planning. Monitoring refers to how progress control is performed. Last but not least, the pair acting and interacting group social & collaboration and information & tool competences.

4 Discussion

Within the planning sphere, the starting condition typically shows 3rd party domination accompanied by mere adoption and offer-oriented selection of goals while at the same time maintaining a degree of inquisitiveness ("stubbornness", "fun in learning", "interest", "inquisitiveness"). Among the necessary triggers we found the ability to explicate intentions as well as objectives and the ability to set priorities to tasks. These need to be developed in order to support the acquisition of the ability to design ones own portfolio and the ability to match formal and personal requirements where necessary.

With respect to reflection, willingness to change own attitudes was identified as the minimal condition to start building a PLE, whereas, at the opposite end of the competence continuum, we found the ability to actively engage in the process of reviewing own (digital) traces, identifying strengths and weaknesses of own performance and thus making creative sense of the past learning experiences. In this case the trigger allowing the passage from one stage to another is the process of "leaving (digital) traces" - producing, publishing, and collecting learning artefacts.

Looking at monitoring, the minimal condition typically includes such 'silver-spoon' learning habits as adopting external evaluation criteria and relying in the evaluation of performance on the more knowledgeable other (typically a teacher). Among the necessary triggers we found the ability to build up criteria for self-evaluation and willingness to "put yourself and your artefacts out [to the wider public] and get feedback by others, or read others contributions for measuring and comparison".

85
These attitudes and abilities are necessary to trigger the acquisition of the ability to control and direct learning progress in an autonomous and disciplined way, and to actively design and direct the relationship with the facilitator(s).

The next area, -- *acting* --, encompasses a set of skills, attitudes, abilities, habits, and knowledge that closely relate to the concepts of Media and Digital Literacies. As a minimal condition for constructing an own PLE a learner needs basic skills in ICT i.e. the ability to operate hard- and software, and the ability to search for, collect and store relevant information. Another very important minimal condition in this sphere is the willingness to try something new. Among the important triggers we identified the following competences (abilities and attitudes) from the interviews: comprehension that information beyond printed media (such as wiki pages, blog entries, or peer comments) have a high value for learning and knowledge construction; ability to deconstruct your PLE i.e. "capability to think about what comprises your personal learning environment"; ability to screen available tools and as a result make an informed selection; and, last but not least, the ability to re-purpose a tool, i.e. perceive affordances emerging from a new context. The above-mentioned abilities and attitudes are an important prerequisite for the acquisition of high levels of digital literacy. These intended high-level outcomes encompass: the ability to assess the quality and reliability of information and the ability to make productive use of technology including matching the right tool for the right job. Furthermore, the construction and maintenance of a tools portfolio is now a competence fitted with a degree of creativity and framed by constantly enriched experiences not only with the tools therein. Whereas the minimal condition requires information gathering skills that are by nature content-centred, the intended outcome is an ability to regulate with respect to both content as well as the learning process.

Shedding light on *interacting*, basic social skills, a social interest, and particularly foreign language skills define the minimal condition. The triggers are mainly characterised by attitude changes: a willingness for exposure needs to be developed -- a willingness such as "[releasing] artefacts to others", "publishing not only reading", or -- more general -- "giving not only taking". Additionally, extended social skills are required to set off the acquisition of the targeted competences. These extended social skills specifically relate to dealing with mediated communication and mediated criticism, often underestimated in their impact on performance. Intended outcomes were identified to be networking competence including the ability to network for feedback, decision competence on when to work in a group (and when not), and negotiation skills. The latter relate to reaching agreement within a group with respect to, e.g., roles, rules, and tools.
5 Conclusions

Within this contribution, we have proposed a draft framework for studying the competence spectrum along five initial dimensions we have elaborated from the given set of interviews. We have partitioned the competence building blocks of skills, abilities, habits, and attitudes into three stages guiding from the situation at the beginning to the intended outcome.

This draft framework sheds light on the competences required for and developed by Mash-UP PLEs (MUPPLEs). Although the framework clearly shows that basic instrumental and informational digital literacy skills are a prerequisite – ‘minimal condition’ – for starting to consciously build a personal learning environment, MUPPLEs can effectively support the development of higher level (strategic) digital and media literacy skills.

The reason for this may be found therein that monolithic applications (such as virtual learning environments, also known as learning management systems) used in education are outdated and do not promote the development of relevant digital literacy skills. It seems that the PLE concept incorporates more recent IT approaches and a more challenging qua truly distributed setting that both require the learner to take a more active role in managing and configuring the involved systems. In short, PLEs can be said to be means to promote digital literacy.

The proposed draft framework and partitioning seems to be valid, but while analysing, we already identified gaps and possibilities for continuing studies. The validity, however, needs to be further studied with the involved stakeholders. We plan to enhance the study with the complement: the competences needed for and developed by facilitators following the PLE approach. Additionally, we plan to verify both frameworks with a quantitative study.

Future work will have to follow up on this to create a more detailed picture of the competence continuum necessary for and motivated by the concept of a personal learning environment.

Acknowledgement

This work has been co-funded by the European Union in the projects Role, TENcompetence, and LTfLL under the Information and Communication Technologies (ICT) theme of the 7th Framework Programme and in the transfer of innovation project Revive under the Leonardo da Vinci sub-programme of the Lifelong Learning Programme.
References


Reflection support using multi-encoded Tag-clouds

Christian Glahn, Marcus Specht, Rob Koper

OTEC, Open University of the Netherlands, Valkenburger Weg 177, 6411AT Heerlen,
The Netherlands
{christian.glahn, marcus.specht, rob.koper}@ou.nl

Abstract. This paper reports on a qualitative study about the application of tag clouds for supporting meta-cognition in self-directed and incidental learning. It analyses the use of a personal tag-cloud visualization of the tags that are used at a public social bookmarking service. The study focuses at the types of meta-cognitive control based on reflection notes of the learners. These notes were analyzed regarding the contents of the reflections as well as regarding their meta-cognitive type. The study has two important outcomes. Firstly, a personal tag cloud can stimulate reflection on the tagging activity of a learner. Secondly, reflecting on the tagging activity is not built into the design of a tag cloud.

Keywords: tagging, visualization, evaluation, reflection, self-directed learning, incidental learning

1 Introduction

This paper focuses on tag clouds that are based on the personal tagging information of a learner. For supporting self-directed and incidental learning tag clouds hold some potential to stimulate reflection on concepts and learning processes, even though such visualisations do not show valid information in terms of approved domain knowledge. A tag cloud can stimulate reflection because the visible tags are meaningful to the learner and allow associations to the actual learning experiences. This paper reports on a qualitative study about using tagging visualisations to stimulate reflection about self-directed or incidental learning activities. This article analyses if the visualisation of a learner's tagging activity in the form of a tag cloud can stimulate meta-cognition of self-organised learners. The present study analyses the types of reflection that are stimulated by a personal tag cloud that is based on a mash-up of different information that is provided by the social bookmarking service delicious.com [3].

2 Background

Tagging stands for applying free form keywords (the tags) to resources that result in user-generated metadata. Tagging is closely related to the developments in the context of the Web2.0 [19]. A number of scientific contributions focus on tagging as a type of
user and community driven creation of meta-data [10, 11], or used tags to improve the accessibility of contents [12, 19]. A number of commercial Web2.0 services supports tagging, because freeform tags provide an easy and flexible way for organizing content and information.

Prior research [2] has argued that labelling and tagging supports meta-cognitive processes in self-regulated learning. Several studies [1, 2, 18] analyzed visualizations of conceptual structures for supporting self-regulated learning in pre-structured knowledge domains. Therefore, the designs of the visualizations were guided by the external learning objectives of the related formal educational context. In these settings the learners could not extend the tagging vocabulary or the conceptual structures.

In contrast to formal education, self-directed learning is defined by a high degree of learner control in weakly structured knowledge domains [16]. This idea of learner control is already incorporated with many Web2.0 services that enable their users to create personal concept structures through tagging rather than replicating predefined concept structures [21]. This has inspired the development of personal learning environments [24].

The concept structures in personal learning environments vary for each user and can range from a few tags to a couple of hundreds. This freedom for the learners requires that visualizations are independently designed from conceptual structures or knowledge domains. Related approaches [4, 13] visualize learner actions that were monitored by the supporting system – the “interaction footprints” [23].

Table 1. Relations between dimensions and factors of situated learning

<table>
<thead>
<tr>
<th></th>
<th>Process</th>
<th>Peers</th>
<th>Event Participation</th>
<th>Concept</th>
<th>World</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lave, 1993</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wenger, 1998</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Presence</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rhythm</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interaction</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Involvement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Connections</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Personal Identity</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communal Identity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relations</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boundaries</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Integration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Community building</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The present paper analyses if tag clouds can stimulate and support reflection in self-directed and incidental learning, which depends on the ability to reflect on actions [5, 20]. Three variations of reflection were identified to be relevant for learning: reflection on action, reflection in action, and reflection for action. The main difference of these reflection types is the occurrence of the reflection. Ertmer & Newby [5] define reflection as an activity that links meta-cognitive knowledge and meta-
cognitive control (self-regulation). The related reflective processes can be clustered into three stages: planning, monitoring, and evaluating.

Lave & Wenger [15] introduced the concept of situated learning. This concept reflects the social dimension of learning. From this perspective learning cannot be reduced to a set of learning events, but requires tight coupling to the related social practices. Further studies highlighted several dimensions and factors of contextualization in learning [16] and contextual support for learning [22]. Table 1 shows the relation between these dimensions and factors. This relation allows identifying, which types of interpretations of the learning processes can be expected for a context dimension. This study focuses at the level of process and concept as the primary context dimensions.

3 ReScope

The present study uses the ReScope system [9]. ReScope is a web-based mash-up for fetching, aggregating, and visualising tagging activities in the social bookmarking service delicious.com [3] and allows learners to annotate and to track their tagging activities. ReScope has provides two tools: a tag cloud visualisation and a tag cloud annotation tool. The ReScope tag cloud provides different perspectives on tagging by combining and contrasting different aspects of tag usage through colour codes and font sizes. With tag cloud annotation learners can track their tag clouds over time: as soon as an annotation is submitted, a snapshot of the current tag cloud is attached to it. When a user clicks at an older annotation, the attached tag cloud is loaded and displayed to the learner. Figure 1 shows an example of the ReScope user interface.

Fig. 1. Screenshot of the ReScope user-interface
The design of the ReScope tag cloud is based on the contextual dimensions concept and process of the contextualization dimensions of situated learning.

1. Concepts: The overall tag usage of each tag is indicated by its font-size.
2. Process: The most recent usage of tags is indicated through color codes.

Because this information is entirely based on the tags that are used by a learner, it is expected that it can serve as an anchor for reflecting about the personal tagging habits and the related knowledge structures.

4 Research objectives and questions

This study has two main objectives: (a) identify if and how tag clouds can stimulate meta-cognition of learners regarding their self-directed and incidental learning on the Web2.0; and (b) identify design factors for the underlying use case for reflection support. Within this scope this study addresses the following questions.

— Does the ReScope tag cloud stimulate meta-cognitive activity?
— Does the reflection on the tag cloud confirm contextual factors suggested by the contextual framework for situated learning?

According to the relations of contextual dimensions and factors, it is expected that the notes can be linked to the context factors rhythm, value, and connection. Rhythm refers to all types of structuring a process. Value covers anything that is related to the added value of concepts and conceptual structures for the learner. The factor connection refers to the structure of the concepts as well as the relation of concepts to resources. The factors “integration” and “community building” are not expected to occur, because ReScope does not visualize connections between peers.

5 Method

The participants were invited through promoted on several mailing lists, web-logs, web-sites of related research projects, and on two European conferences in the domain of computer assisted learning. The participation was entirely voluntarily. The interface of the ReScope system included information texts that asked the participants to write down their thoughts about their reception of the tag cloud in the notes field and to label important words with a hash (#). This technique is also known as hash tagging.

Nine weeks after the initial announcement of the system the system's usage has been analyzed. “Noisy” statements like as “hello” or “xxx” were removed prior to the analysis. The remaining notes were labeled with the three types of meta-cognitive control of Ertmer & Newby [5]: evaluate, plan, and monitor. Additionally to this model based labeling, all notes were clustered based on the contained information using content based labeling based regarding the context factors of situated learning.
6 Results

Over the period of nine weeks, 76 users have registered with ReScope. According to the domains of the registered e-mail addresses, participants from 13 countries were identified. In average, the respondents were 36 years old; the youngest participant was 22 and the oldest was 58 years old. The majority looks at the official tag cloud at delicious.com (91%), although 73.5% of the respondents state that they look at their tag cloud only sometimes or rarely. Nine participants left 43 notes about their tag cloud. After cleaning the noise from the notes, 35 notes remained that describe reflective activity of their authors.

The first notes of five users reflect their surprise over the structure of their personal tag cloud and expressed the need to adjust the tags. An example for these notes is given by User A.

“I have too many occurrences of singleton tags #clean-up”.

The note of User A also shows the use of a hash tag. This feature has been used by six participants in 14 notes. The provided hash tags were beneficial for labeling the notes for the analysis. The following two examples illustrate the use of hash tags. In the first note, User F stated.

“need to cut out #duplicate_tags”

A more complex example of hash tags is given by the statement of User D.

“Writing on my #blog (++space) I realised I hadn’t used #tag podcast very consistently. #consistency important, so I #edited the post.”

All messages were labeled according the three types of reflection for meta-cognitive control: evaluate, plan, and monitor. One note was labeled with evaluation and planning because the note stated first the result of a self-evaluation and then defined an objective for future tagging. 17 notes are labeled as evaluative reflection, six notes as planning, and 13 notes as monitoring.

The results from the model centric labeling show that all three types of reflections were identified with the notes (see Table 2). The majority of the notes were labeled as evaluative reflection. A fifth of all notes and two fifths of those that were tagged as evaluative reflection referred to cleaning the tagging structure.

Additionally a content based labeling has done on all annotations. After this procedure a note could have many labels assigned. This step brought up three key categories: 23 notes were reflections about the semantic structure of the tag cloud, six notes were reflections on organizing the tagging process or on aligning it with other tasks, and nine notes were reflections on system features. Furthermore, two notes were labeled as reflections on social interaction and one note was labeled as a personal remark. Four notes were classified as follow-ups to prior notes that marked steps in a process. Within the notes on the semantic structure, three sub-categories were identified. The participants reflected in seven notes on the need of cleaning up the tag structure, in eight notes about the consistency of how the tags are used, in two notes about emerging tags.
7 Discussion

The present study focused on the potential of tag clouds for supporting reflection on the personal tagging of self-directed learners. It combines and extends the prior research on visualizations of learner activity based on knowledge domain independent information [6, 7] as well as the research on using tags for identifying learner interests in open ended environments [8, 9].

The results show that already the rather simple visualization stimulate reflection are conforming the models of meta-cognitive control. The notes of the content based labeling are equally distributed across all types of reflection (Table 2). This suggests that the tag cloud supports meta-cognitive processes in self-directed learning of all levels similarly. Based on the present results it can be argued that tag cloud visualizations can serve as a reflection support tool for self-directed learners.

Table 2. Results of model and content based labeling

<table>
<thead>
<tr>
<th>feature</th>
<th>process related</th>
<th>semantic structure</th>
<th>consistency</th>
<th>clean-up</th>
<th>emerging topics</th>
<th>general reasoning</th>
</tr>
</thead>
<tbody>
<tr>
<td>evaluate</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>plan</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>monitor</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

The results of the content labeling can be interpreted as instances of the context factors for situated learning. By mapping the identified labels on the context factors, it becomes possible to address the second research question. The design of the tag cloud focused at the context dimensions process and concept. Therefore, it is expected that the participants' notes are related to the context factors rhythm, value, and connection (Table 1). Notes that were labeled only as semantic structure but not to any of the sub-categories were mapped to the connection factor, because these notes reflect the relations between the tags and the relations between tags and resources. Notes of the categories process related and emerging topics were mapped to the rhythm factor.

Table 3. Relations of labeling after remapping content labels to context factors

<table>
<thead>
<tr>
<th>rhythm</th>
<th>value</th>
<th>connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>evaluate</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>plan</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>monitor</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Total</td>
<td>9</td>
<td>15</td>
</tr>
</tbody>
</table>

Finally, the notes of the categories clean-up and consistency were mapped to the value factor, because these notes refer to the value of the tags for the personal information management. Table 3 shows the results after remapping of the contextual
factors. The results show that the reflections related to the three factors were almost equally stimulated by the tag cloud. This finding supports the expectations regarding the relevance of the contextual factors for reflection.

8 Conclusions

Reflection is the active reasoning that has been described as the learner's assessment and validation of experiences in problem solving and social interaction against conceptual structures and strategies. This process is part of developing consciousness about the relations between actions, beliefs, and concepts. Therefore, reflection is an important meta-cognitive process related to learning. This article analyzed if a tag cloud visualization of a learner's free form tags can stimulate such meta-cognitive processes. This study has two important results.

1. A personal tag cloud can stimulate reflection on the tagging activity of a learner.
2. The concepts of situated learning can be applied for developing tools for supporting self-directed learners.

The findings of this study indicate that targeted solutions for supporting meta-cognition do not need to be dependent on pre-structured domain knowledge. This opens opportunities for developing new forms of supporting self-directed and incidental learning in personal learning environments.

9 References


Conceptual Design of EduFeedr — an Educationally Enhanced Mash-up Tool for Agora Courses

Hans Põldoja¹, Mart Laanpere¹

¹ Tallinn University, Institute of Informatics, Narva mnt 25,
10120 Tallinn, Estonia
{Hans.Poldoja, Mart.Laanpere}@tlu.ee

Abstract. The use of blogs in online courses has become increasingly popular. Current RSS feed readers lack special features to monitor the process of collaborative knowledge building in these agora type of courses. In this paper we present the conceptual design of an educationally enhanced feed reader and mash-up tool that is work-in-progress.

Keywords: mash-up learning environments, feed readers, scenario-based design

1 Introduction

Increasing number of educational institutions are making their courses openly available in the Web. However, open course could be a confusing term because openness can have several meanings: open access to static content such as in MIT OpenCourseWare, use of an open learning environment based on Web 2.0 tools or a course in traditional LMS with an open registration. In this paper we are focusing on specific kind of open courses that share all of the following three criteria: (1) openly accessible content, (2) open personal learning environment, and (3) free and open registration for participation. We propose to use a new term — agora course — while referring to courses of this kind. Agora is a Greek word for a place for assembly or an ancient marketplace.

In spring 2008 the author of this paper was teaching an agora course in University of Art and Design Helsinki. On that course Wikiversity was used as a central component of the learning environment [1]. Other tools that were used in the course contained teachers’ blog in Wordpress.com, students’ blogs on various blogging platforms, social bookmarking service (Delicious) and Web 2.0 environments for sharing content that was created by the participants (LeMill¹, photo sharing websites, podcast hosting websites and video sharing websites). The course started with more than 50 participants who signed up on a wiki page. Number of people who sign up for the open course may grow rapidly if the course receives attention in the blogosphere. For instance, more than 1200 people signed up for an open course Connectivism and Connective Knowledge in summer 2008 [2].

¹ LeMill. http://lemill.net
While setting up a course wiki (e.g. at the Wikiversity portal) is a simple task, it
does not cover all the functionalities needed for smooth and effortless management of
teaching and learning activities. This paper addresses the challenges of designing one
of the central components of a mash-up personal learning environment for agora
courses: an educationally enhanced mash-up tool that supports planning, monitoring
and analyzing learning activities which cross the borders of different Web 2.0
applications.

2 Aggregating News Feeds in Mash-up Learning Environments

Large number of participants who all post in their personal blogs (which, in most
cases, are using different hosting providers and blog software) and use various Web
2.0 environments for other type of content creates a challenge for a facilitator: how to
reduce the overhead work while trying to follow the activities that take place on the
course and to provide feedback for all the participants.

Currently the authors have used Pageflakes online feed reader for creating shared
pages that display all the blog posts and comments. These pages are useful to get a
quick overview but the recent changes are not specially emphasized. Therefore a
standard desktop RSS reader was also used. During these courses it came out that
standard feed readers lack several features that would be important for online courses.
For example in RSS readers typically new posts are emphasized, but in a course with
many participants it is equally important to notice when somebody is left behind. It is
also important for the facilitator to see in one glance the “big picture” of the activities
taking place on the course, especially the feedback given by one student to another (in
order to monitor the intensity of collaborative knowledge building). These special
features have been always present in traditional learning management systems, but if
we want to keep our agora courses as open as possible, we should allow our students
to use their own personal learning environments.

There have been some studies and developments in the similar direction such as
MUPPLE [3], simple aggregating mash-up The Wire [4] and knowledge discovery
application PALADIN [5], but none of these are addressing exactly the needs.

3 Conceptual Design of EduFeedr

Our study is based on the design-based research methodology [6], involving iterative
scenario-based design [7] and lightweight prototyping [8].

In order to clarify the design requirements a lightweight prototype was built using
Bento personal database application for Mac OS X. All the blog posts and comments
that were written by the participants were stored in Bento database in webarchive
format. It was possible to browse posts and comments by the participant, write short
notes about the blog posts and add ratings. The prototype was used in a course that
had quite detailed assessment scheme. Final scores for all the students were calculated
automatically based on the ratings. Some students decided to delete their blogs soon
after the course was over. Complete archive of course discussions was still available
in the database. The main limitation of Bento database was lack of RSS reader. All the blog posts had to be saved manually. Because of these limitations it was decided to design an online tool that would function both as a feed reader and database for managing course discussions.

This new tool called EduFeedr had the following initial design requirements: (1) it should aggregate information from all major blogging platforms using open standards (RSS, Atom, trackback, pingback); (2) it should not require any special plug-in on the student blogs; (3) the scope of EduFeedr is limited with aggregating and annotating the feeds from both teacher’s and students’ PLE’s and visualizing the process of knowledge building; (4) only teacher has an user account in EduFeedr, which allows her to modify the EduFeedr settings; (5) anyone has read access to aggregated course content.

In the second iteration of the design we wrote a set of scenarios that describe the main features of EduFeedr. Six scenarios covered typical use cases such as setting up the course, writing a blog post, browsing the blog posts, writing comments and archiving the course. The scenarios were published in a wiki where other people were able to post their comments. An example scenario is presented in figure 1.

John has been using EduFeedr for a few weeks. For him the most exiting feature is the way how connections between the blogs are presented. EduFeedr has a visualization where all the blogs are displayed as nodes. Lines between the nodes show the links between the blog posts. All the students have linked to the course blog. Some of the student blogs have a lot of connections while others have not been so active.

It is possible to switch on a different view and see who has commented which blog. This time John finds out that some student blogs have actually more comments than his blog. The same information is also displayed as a table where it is easy to see how many pingbacks and comments each participant has made. EduFeedr has also aggregated all the comments. It means that John can see all comments that one student has made on a same page without visiting all the blogs. This will save him a lot of time, because commenting is part of his grading scheme and students get points for that.

Fig. 1. Example scenario: Exploring the connections between student blogs

In order to start using the EduFeedr a course must be created by the teacher. The students can use any blogging platform that they like. It is suggested to create a new category for course related blog posts or a special blog for the course. Instead of adding their blog addresses to the wiki page the students will submit their blogs on the course page in EduFeedr. The facilitator can also specify tags that have a specific meaning. For example it is possible to make all blog posts with a tag “urgent” to stand out in the feed.

One of the complicated aspects of blog-based courses is connecting students’ blog posts with the assignments. One approach is to connect blog posts with assignments by the date when the post was published. This does not require any special input from the student but unfortunately it would work only when all students submit their posts in time. Another approach is to agree a special tag for each assignment. Our experience with using tags shows that it is quite common to misspell or leave out the tag. In EduFeedr scenarios we decided to use a third approach. Students are asked to

---

include a link to the original assignment in their blog post. This is a common blogging practice to refer to another blog. Course blog will receive a notification comment using trackback or pingback method. EduFeedr can connect assignments with blog posts based on these notifications. This approach has also some limitations. Blogger blogging platform do not send trackback or pingback notifications to other blogs. That can be solved by parsing all student blogs with Trackback ‘em All script.

4 Conclusions and Future Work

This paper described the work that is still in progress. The next step is to choose a suitable open source platform to be used for developing and validating EduFeedr prototype. Potential candidates are POSH, rawdog and Tornado.

Acknowledgments. This study was funded by MER targeted research grant No. 0130159s08.

References


4 POSH. http://sourceforge.net/projects/posh/
6 Tornado. http://www.tornadoweb.org
RDFa versus Microformats: Exploring the Potential for Semantic Interoperability of Mash-up Personal Learning Environments

Vladimir Tomberg, Mart Laanpere
Tallinn University, Narva mnt. 25, 10120 Tallinn, Estonia
vtomberg@tlu.ee, mart.laanpere@tlu.ee

Abstract. This paper addresses the possibilities for increasing semantic interoperability of mash-up learning environments through the use of automatically processed metadata associated with both learning resources and learning process. We analyze and compare potential of two competing technologies for this purpose: microformats and RDFa.

1 Introduction

Mash-up Personal Learning Environments have become a fast developing trend in the world of technology-enhanced learning, partly because of their flexibility and lightweight integration features. Although it is quite easy to aggregate the RSS feeds from the blogs of learners, it is more difficult to get an overview of course and its learning activities. A course is not just a syllabus, it also involves various dynamic processes that can be described in many aspects. The course always has certain learning goals, a schedule that consists learning activities (assignments, discussions), registered participants like teachers and students, and different types of resources. It would be useful, if we would be able to extract such information also from mash-up personal learning environments (just like it can be done in traditional Learning Management Systems) and allow exchanging it between the course participants.

Today for semantic tagging of Web content in general and learning content as special case various technologies are used. But there are no tools and ways exist for semantic annotation of learning process that takes place in a distributed network of mash-up personal learning environments. The main aim of this paper is to compare possible ways for embedding the rich metadata about the course into mash-up personal learning environments.

The main format of presenting information in mash-up PLEs is an (X)HTML document; so any necessary metadata should be embedded into it. Because (X)HTML syntaxes were not designed for carrying the semantic data, different enhancement technologies were introduced in the past. We try to examine two of them that seem for us as much appropriate for the aims mentioned above — microformats and RDFa. The results of such inquiry could be useful in first turn for designers of mash-up learning environments, because there are no common established issues that can be used for semantic annotation of learning process.
2 Scenario

Let us examine a typical usage scenario of using mash-up personal learning environments, involving exchange of semantic metadata associated with learning-related content and also the process of learning. As a starting point for initiating a new course, the teacher publishes an announcement using a Web application (blog, wiki, forum or personal Web site). This announcement contains (meta)data about the course syllabus, pre-requisite and target competencies, amount of credits, dates for start and end of the course, the criteria and form of a final assessment, contact information of teachers and other participants. Web servers by means of mash-ups will bring this information to potentially interested learners. It arrives to them through RSS channels, Web services and/or an e-mail subscription. After a learner enrolls him- or herself to a course, information about him/her is added to a course database.

The teacher attaches semantic significance to learning materials placed in a network. The learners use semantic search for finding additional learning resources. Besides, by means of metadata additional organizational possibilities appear. All participants of a course, learning materials, activities of the teacher and learners can be connected among themselves by means of semantic relations that give the possibility to trace progress of learners in real time or for reception of reports. In this case the classical learning system can be implemented as the thin client which role to search and analyze semantic relations. Problems of indexing and storage of data and representation of information take upon oneself distributed servers. The role of learning system at such scenario is reduced to aggregation and the analysis of the distributed data only.

Using only the metadata which is intended for the learning materials description will not be sufficient for current scenario. The metadata by means of which is possible to describe a course is required for this purpose. In wide understanding the course it is not a static product, it is a process which after starting occurs throughout time. Also the course is the community of people that work together. The curriculum can be implemented as dozens copies of course, but each of these copies, besides a set of the identical information will have also the own unique information associated with various resources, including the human resources. Usual course’s metadata contains values that possibly will not change, for example the course name, amount of credits or form of the assessment. At the same time the course expiration date, amount of the learners registered on a course, names of teachers and a venue will change every time as new instance of course will initiate.

Thus, using a special semantic markup for courses, the teacher has possibility to constantly update the course information during time. The teacher assigns lectures, announces an assessment and evaluates learners. Learners thus constantly have the fresh information on everything that happens on a course. After finishing of a course the information about it as about a logical unit also can be demanded for the analysis and the reporting. We will consider two technologies available today that allow implementing semantic description of educational process occurring in a Web environment.

3 Microformats

Microformats are defined as “a way of adding simple markup to human-readable data items such as events, contact details or locations, on Web pages, so that the
information in them can be extracted by software and indexed, searched for, saved, cross-referenced or combined’ [1].

Microformats allow to use for metadata storing purpose the standard (X)HTML attributes 'class', 'rel' and in a case with microformat VoteLinks as well 'rev'. The Most important advantage of microformats consists in that they use existing HTML syntax, thus not changing it at all. The HTML text with correctly used microformats always will pass a validation. Most of the Web browsers simply ignore the presence of microformats in the HTML code. Recently some developers of browsers already had announced further microformats compatibility, though at the current moment visualization of microformats is possible only by means of additional add-ons or coding of special page behavior, e.g. by means of JavaScript.

Microformats are categorised into two groups: elementary and compound microformats. Elementary microformats like geo or rel-license, are intended for the describing of small, individual things and concepts. They can also be used as building blocks for compound microformats. Compound microformats have complex nested structures. Examples of compound microformats are hCalendar, hCard that precisely reflect data of standardized iCalendar and vCard.

The basic advantages of microformats are their simplicity of implementation, a consistency with existing standards and broad support of developers. Possibility to place in the text additional information which without unnecessary expenses can be transferred to an address book, a calendar or an electronic map attracts users. These possibilities undoubtedly add accessibility to data and make the web friendlier to the user. Support of microformats through special extensions is implemented today almost in all popular web browsers. In comparison with other solutions microformats seriously facilitate a life to developers. They should not implement specific behavior for Web clients again if someone already implemented it. Microformats are not standardized by any standardization agency, but they are well specified and widely known. If one developer writes the program for processing of microformats it will process such microformats from any sources.

The vocabulary of microformats is constantly updated. Unlike well-known standards microformats have no versions, they are being developed continuously. People responsible for uniformity of microformats try to maintain the vocabulary of microformats in as much as possible compact condition. In most cases they advise instead of introduction of a new microformat to try to use already existing microformat. At the same time in many cases microformats’ vocabulary is not sufficient for the description of all necessary data. In such case content creators appear in front of a deadlock.

Biggest drawback of microformats is that they have no ontologies, formal descriptions or schemes. The vocabulary can be checked up manually under specifications, but there is no way of automatic check. The existing values of 'rel' and 'class' attributes can to be not coinciding with the vocabulary of microformats, but thus HTML code remains completely valid [2]. In HTML code any value for these attributes can be used, but not in microformats’ case.

4 RDFa

RDFa it is the syntax standardized by W3C. RDFa is an acronym from RDF in Attributes; it uses the mechanism of implementation of metadata similar to microformats in XHTML attributes. Instead of three attributes of microformats RDFa
uses ten; sometimes at first sight it is difficult to distinguish code of RDFa from microformats one. For example:

```html
<a rel="cc:license" href="http://creativecommons.org/licenses/by-nc-nd/3.0/">
Creative Commons License
</a>
```

Difference of RDFa in the given example is syntax of 'rel' attribute value. A 'cc:' prefix specifies namespace, in this case Creative Commons. The prefix allows to RDFa to describe such concepts which by means of microformats it is hard to describe, especially because of the limited vocabulary of microformats.

The fundamental advantage of RDFa that with its help it is possible to use different namespaces in one document, for example 'dc:' as Dublin Core and 'cc:' as Creative Commons simultaneously. RDFa allows combining, remixing, and extending existing vocabularies easily [3].

Using of RDFa assumes for developers also some considerable restrictions. It is possible to apply RDFa only to XHTML version 2, not to HTML. XHTML version 1.1 also can be used with RDFa in practice but with considerable limitations [4]. Prospects of use RDFa together with HTML (not XHTML) do not looks as promising at present. On the one hand attempt to introduce RDFa code in HTML will lead to situation, where Web browsers can process such page without any problems. However the validation of such HTML+RDFa page will be impossible, as in HTML syntax is absent necessary for RDFa attributes. In addition RDFa code can be casually damaged at attempt to pass HTML through Tidy or other code cleaning tool. Such tools will simply break embedded RDFa semantics [5]. In consideration of mentioned above using RDFa in HTML should have no trust from developers.

Support of RDFa syntax likely is not planned to introduce in following version HTML 5 also [4], though advocates of a RDFa do not lose till now hope that WHATWG will pay the attention to problems associated with semantics [6].

Other barrier for fast RDFa introduction is using in it a new syntax for Uniform Resource Identifier (URI). Instead of traditional URI a new CURIE syntax is used in RDFa. CURIE simplifies work with links on URI and shortens a code. CURIE is W3C Candidate Recommendation [7]; it is intended for use in SPARQL, RDFa and XHTML 2. However, CURIE technology is still in its early phase of development.

5 Microdata — a new player

At the moment of writing this paper (July 2009) important changes in situation around standards have happened. The W3C announced that ‘XHTML 2 Working Group expected to stop work end of 2009, W3C to increase resources on HTML 5’ [http://www.w3.org/News/2009#item119]. Obviously reason for this decision was changes in relationship between W3C and WHATWG as browser’s business representative. W3C has changed its attitude to HTML 5 from full rejection to unstinting support.

Dozens discussions in the W3C Mailing lists and blogs related to standards have lead to much unexpected result. Ian Hickson, a leader of HTML 5 working group finally has agreed to arguments of his opponents and has proposed a Microdata — completely new semantic syntax for HTML 5 [8]. By its nature Microdata is a certain kind of hybrid made from microformats, RDFa and HTML 5 syntaxes. It is based at three new HTML 5 attributes: "item", "itemprop" and "subject"; at their base RDF triples can be generated.
At the moment of writing of this paper a discussion about strengths and weaknesses of Microdata is continuing. Advocates of Microdata, for their part, are proposing analysis of use cases that describe advantages of Microdata compared to RDFa [9]. Advocates of RDFa in turn publish their own RDFa at the HTML 5 specification’s proposal [10] and express optimism about future perspectives of RDFa.

Situation with RDFa is remaining to be very unclear. As W3C says, work on XHTML 2.0 and CURIE will stop, although both of these unimplemented standards very close related to RDFa. In the same time W3C says also: ‘Whether and how to include RDFa into HTML 5 is an open question on which we expect further discussion from the community’ [11]. Uncertainty of RDFa’s perspectives and appearance of new player Microdata only complicated tasks for developers and will slow down widespread use of semantic tools on the Web platform.

Because Microdata for today is not established and well defined yet, we do not consider it in this paper.

6 Microformats vs. RDFa on the semantic level

After outlining tech differences of Microformats and RDFa we try to define a differences between two technologies in semantic aspects. Both of them have own historical background, both are driven by forces from very different sources, and especially important, these technologies have very different perspectives.

From the point of view of data description the principal difference between microformats and RDFa consists that microformats can use only one vocabulary. This vocabulary is being approved by community and cannot be modified by the separate developers. Actually it is possible to concede such independent modification in practice, but it should be considered as violation of microformats’ principles, so we not examine such case. RDFa vocabularies on the contrary are not limited by a specific data set [12]; they are developed separately in different places. In connection with the aforesaid it becomes obvious that implementation of semantic data in big scales is easier with using of microformats. The probability that software will easy manage, define and process any syntax of microformats is much higher than probability of recognition of RDFa, just because this vocabulary is small, well known, and is just one.

At present there are some implementations of add-ons for Firefox and Internet Explorer browsers exist which are able to recognize both formats. But only vocabulary of microformats is well known for everyone who interested to work with them.

Both technologies can be considered as from side of supporting of end users, as from side of machine processing for indexing and mashups needs. The developers already have many possibilities to implement practical functions for end users using microformats today. For example, if Operator or Tails Export add-ons for Firefox have a detection of hCard microformat’s presence, they suggest to the user an export of web-content to a file with .vcf extension or a method for adding the information to personal address book. Detection of hCalendar microformat calls a suggestion from browser to export the content to a file with .ics extension or place data from web page directly to the calendar software of the user. Tails Export add-on if meet a web-page with Rel-licence microformat can show to the user the specific information about this license, or in a case if there is microformat Geo presented, the software suggests an
allocating of the address on the Google Maps. All above listed examples are finished and working implementations of possibilities of microformats use for end users.

With RDFa the situation differs radically. Some add-ons for Firefox declaring ability to work with RDFa, but today they are able only to find out RDFa code on the web-page. A maximum that a browser with RDFa code can make, it is a retrieving RDF triplets from source code and sending the user on a special page with the RDF triplets’ interpreter. It is obvious, that such services can bring some advantage for developers, but have no sense for end users, and actually there is nothing especially useful to the end user in RDFa.

The implementation of machine processing on a global scale is much easier too with the vocabulary of microformats than RDFa’s because vocabulary of microformats is uniform and developed centrally. The software from different vendors should be able to process microformats of one type equally. The set of microformats’ types is limited by community therefore it is possible to state high enough level of interoperability between the different software intended for processing of microformats.

At the same time uniform vocabulary of microformats has also drawbacks. Developers continually face a deficiency of the vocabulary, with absence of new microformats which are necessary to cover continually arising new requirements. Other possible problem consists in a non-typical way of microformats’ development, because unlike traditional standards, today it does not assume any logical completion. Specifications of existing microformats, their syntax and the processing rules are not described in schemas, they are only can be searched manually through a web [13]. Such approach obviously does not promote fast updating of the software after appearance of new microformats. The majority of developers simply learns nothing about new developments if does not monitor a web site microformats.org on one's own.

7 Educational application

Above mentioned semantic characteristics of technologies under comparison set quite different limits to their educational use. We try to examine these limits closer now.

The vocabulary of microformats is laconic and almost perfect. We can solve many tasks by means of the microformats’ vocabulary if it is necessary to describe activities concerning a course. Developers of microformats suggest using actively a nesting of microformats, and by means of that it is possible a semantic marking of rich data structures.

Let's consider some examples of possible using of standard vocabulary of microformats for the course description. By referring to use case described above, lets examine the situation, where the teacher announces a course. We can use for the description of this course a microformat hCard. This microformat allows naming a course using a class “organization-name”. Also with the help of hCard we can specify geographic location of the course. With the help of hCalendar microformat we can announce lessons, specify their duration and a venue. The learner can then automatically import a schedule of lessons to their personal calendars.

Similarly we will supply with hCard microformats all data about the teacher and learners who take part in a course. By means of the address information learners can find each other and communicate among themselves and with the teacher online. We can also define a difference between them by means of a ‘role’ class.
Shared knowledge building assumes the organization of workgroups from learners. Here we at the first time face shortage of microformats’ vocabulary. If a necessity to specify the relation between participants of a course (teacher / learner) or participants of working group (senior of group, rank-and-file member of group) it would be logical to use XFN microformat which is created to show relations between people. However the best possible solution (which is thus far not ideal) that can be found in XFN are values of 'rel' attribute “co-resident” and “colleague”. It is certainly not the best method to show the exact relationships in groups. Values of attribute “parent” and “child” for such situation not so good suits also, as in a context of course have ambiguous sounding.

This example well illustrates problems of implementation of specific ontologies by means of microformats. On a web-site microformats.org there are special pages on which the community discusses new microformats for education. There even the prototype of the future microformat hCourse is mentioned. Unfortunately, planned functionality of it would more suit for the description of the curriculum than for description of the course process. In existing attempts to implement the description of the course, the course often is considered as the set of certain curriculum data, learning materials and other accompanying information. For example such data can be rooms where the course will be spent, teachers’ names, final examination’s date and so forth. The problem consists that the information on a course is considered first of all as the marketing tool by means of which is possible to inform a buyer about university products. However such information will be almost useless after the course will begin. We wish to notice that course is not only static set of the information. It is also selection of activities into that many people and resources are involved.

Table 1. Summative comparison of Microformats and RDFa. Better features marked with gray background.

<table>
<thead>
<tr>
<th></th>
<th>Microformats</th>
<th>RDFa</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Technological properties</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Can be applied to</td>
<td>HTML, XHTML</td>
<td>In current state for XHTML 2 and limited for XHTML 1</td>
</tr>
<tr>
<td>Have useful implementations for end user today</td>
<td>Yes, different add-ons for browsers allow that</td>
<td>Not very useful for end user directly</td>
</tr>
<tr>
<td>Can be used in mash-ups</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Practice using for semantic indexing examples</td>
<td>Google and Yahoo now indexing such microformats as hCard and hReview</td>
<td>Google, US Government Website and Slideshare use RDFa</td>
</tr>
<tr>
<td>Standardized</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Drawbacks</td>
<td>It is very difficult to use one limited dictionary for all purposes. Because of absence of standards support from developers is more enthusiastic than industrial</td>
<td>Because of finishing of XHTML2 development the perspectives of RDFa are very unclear at the current moment</td>
</tr>
<tr>
<td><strong>Semantic properties</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vocabulary</td>
<td>Only one</td>
<td>Unlimited amount is possible</td>
</tr>
<tr>
<td>Vocabulary can be extended</td>
<td>Yes, but only through community</td>
<td>Yes, freely</td>
</tr>
<tr>
<td>Interoperability level</td>
<td>High</td>
<td>Possible only if common vocabularies are used</td>
</tr>
<tr>
<td>Possibilities to add semantic value to data</td>
<td>Average</td>
<td>High</td>
</tr>
<tr>
<td><strong>Application for educational needs</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Possible semantic data use</td>
<td>Course’s membership via hCard Learner and Teacher information via hCard Members relations via XFN Course events via hCalendar</td>
<td>Can use any semantic vocabulary that suitable for course description.</td>
</tr>
</tbody>
</table>

At RDFa the situation with vocabularies is much better. Possibility simultaneously to use at once such popular vocabularies as Creative Commons, Dublin Core, XCRi and other removes both problems related to description of a course resources and to description of activities. In case of absence necessary metadata vocabulary it is
possible to create any schemes by oneself. However it is necessary to mean, that interoperability of system will decrease proportionally to appearance of new vocabularies, because any software cannot be ready to right process all possible vocabularies in the world.

A general outline of microformats and RDFa comparing results is shown in table 1.

5 Conclusion

So which technology is more suitable if we need to exchange easily the rich metadata about the course and its learning activities between several mash-up personal learning environments? The answer is depending on our choice between the universality of language and simplicity of adaptation of the selected annotation technology. If developers of microformats will start paying more attention to educational possibilities of microformats (and they probably will), it is possible in the near future we will see new microformats that are powerful enough and simple for adaptation. If we decide to implement the RDFa approach, there is still a lot of work to do. We will harvest on much smaller field, than in a case with microformats. The output however will be much more flexible and semantically rich than with microformats.

6 Acknowledgements

This study was funded by ESF grant 7663 and MER targeted research 0130159s08.

References

Interoperability issues for formal authoring processes, community efforts, and the creation of mashup PLE

Roland Klemke¹, Birgit Schmitz²

¹ Open University of The Netherlands
CELSTEC
Valkenburgerweg 177, 6419 AT Heerlen (NL).
roland.klemke@ou.nl

² Humance AG
Goebenstraße 10-12, 50672 Köln.
Germany
bsc@humance.de

Abstract. This paper compares three different scenarios towards the creation of learning offers and looks at corresponding interoperability issues involved. In outlining use cases that demonstrate three possible authoring/learning scenarios, restrictive and supporting aspects become obvious and eventually lead to the deduction of prerequisites and requirements that are vital for the setting up of an effective (personal) learning environment.

Keywords: interoperability, content creation process, personal learning environments

1 Introduction

In professional formal learning media production, standardised process models and interoperability standards are the results of the trade-off between creative collaboration and organisational quality requirements.

Web 2.0 communities as well as efforts towards open educational resources are built around the notion of voluntary, free and open collaboration. Interoperability issues arise when different communities meet. However, in PLE research, little has been said about collaboration and interoperability issues despite a strong trend towards the notion of personal learning environments as personalized and individualized instruments [5].

In a previous workshop [3], we compared for three different settings the content creation process and the possible outcome in regard to learning possibilities and interoperability issues. Therefore three different groups had to work on the same task (to collaboratively create an opportunity for learning a language) using different approaches (formal authoring of content, community approach, setting up a PLE). With this paper, we want to present some results of that workshop.
2 Setting up content

Formal content production processes, community based Web 2.0 offers and the creation of mashup PLEs represent three different approaches to produce units of learning in the sense of a contextualized, self-contained unit of education or training, such as a concrete course, module or lesson.

2.1 Creating content with a formal authoring tool

Setting up a learning offer with a formal authoring tool very often refers to scenarios in which professional authors collaboratively create content i.e. units of learning. They collaborate in order to optimize the result by exchanging information and knowledge. Collaborative authoring implies simultaneous (real-time) and asynchronous collaboration. Authors and learners are distinct groups.

This scenario is usually applied in formal learning settings by companies, educational institutions, etc. Content delivered is usually considered to be “final” or “approved”.

Authoring tools are commonly used to produce learning content for a defined learning target, a specific learning group, etc. The content has a defined scope and extent. Its creation and delivery follows a formal mostly standardised process.

To produce learning content, authoring tools have to be interoperable along the production chain. Most authoring systems support certain standards like SCORM [6], LOM [7], AICC [8], IMS LD [9], IMS QTI [10], etc. Content set up according to these standards runs on all systems supporting these standards.

2.2 Content creation with community based Web2.0 Tools

Creating content with a community based Web2.0 tool means creating materials and providing tools for learning and also arranging them in a learning environment that encourages individual and social activities.

Creating a learning community content does not mainly focus on informing or imparting knowledge but also serves as a starting point for individual reflections as well as discussions with others. It is “alive” and changes rapidly according to community requirements. Members/learners contribute as well as consume.

The content creation process and accordingly the content itself do not necessarily meet formal (learning) requirements. Social software systems are mostly self-organized and characterized by communication and collaboration. Therefore they especially suit phases of informal learning [11].

Still, this scenario involves the notion of content as a medium given from the collaborative authoring team to the community of learners (which may overlap) and different phases of its creation may be distinguished. Also, when setting up content, interoperability issues regarding the use of tools, platforms, etc. arise and have been addressed with standards and initiatives like OpenId [12], OpenSocial [13], or OAI [14].
2.3 "Authoring" a mashup PLE

A mashup or widget/gadget environment as part of a Personal Learning Environment takes into consideration all different sorts of learning activities (web-based and non web-based). As an individual setting it is created on a widget platform according to one’s own needs and can serve different purposes such as information, learning, gaming, etc.

To create a mashup learning environment it is necessary to collect and choose available and suitable widgets as well as different tools for communicating. No formal (learning) requirements restrict the choice; no quality assurance/control is applied except for the personal assessment of suitability.

This scenario does not involve a distinction of authoring team and learning community: the author (or creator) of the PLE does it for his/her own purpose. Consequently, a PLE is not primarily about content, but rather about tools, services, and resources. In this context interoperability issues between the different components of the PLE arise [4]. As a result to this, widget standards (W3C widget Requirements, Open Ajax Alliance’s Gadget) emerge in order to drive interoperability. However, standards that reflect learning needs in widget environments are still missing.

3 Conclusion

Table 1 compares the scenarios concerning collaboration issues, interoperability focus, standards and summarises the key settings for each approach in the conclusion.

<table>
<thead>
<tr>
<th></th>
<th>Formal authoring tool</th>
<th>Community based Web2.0</th>
<th>Mashup PLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaboration</td>
<td>Formally defined pro-</td>
<td>Self-forming teams,</td>
<td>Learners may</td>
</tr>
<tr>
<td></td>
<td>cesses and roles,</td>
<td>dynamic collaboration by</td>
<td>communicate on</td>
</tr>
<tr>
<td></td>
<td>collaboration takes place</td>
<td>volunteers</td>
<td>efficient and</td>
</tr>
<tr>
<td></td>
<td>along production process</td>
<td></td>
<td>effective constituents</td>
</tr>
<tr>
<td>Interop-</td>
<td>Between Content and</td>
<td>Ease cross community</td>
<td>Between components</td>
</tr>
<tr>
<td>erability</td>
<td>Learning Environment</td>
<td>work</td>
<td>of the PLE</td>
</tr>
<tr>
<td>Standards</td>
<td>Well-defined standards</td>
<td>Standards emerge (openID,</td>
<td>Interoperability issues not</td>
</tr>
<tr>
<td></td>
<td>available (SCORM,</td>
<td>OpenSocial, OAI)</td>
<td>yet systematically</td>
</tr>
<tr>
<td></td>
<td>AICC, IMS LD)</td>
<td></td>
<td>researched</td>
</tr>
<tr>
<td>Conclusion</td>
<td>Collaboration is an</td>
<td>Collaboration is a key</td>
<td>Collaboration is an</td>
</tr>
<tr>
<td></td>
<td>efficiency factor</td>
<td>success factor.</td>
<td>add-on feature in PLE. Key</td>
</tr>
<tr>
<td></td>
<td>supported by</td>
<td>Interoperability</td>
<td>benefits (highly tailored</td>
</tr>
<tr>
<td></td>
<td>interoperability</td>
<td>standards ease cross</td>
<td>to individual needs) come</td>
</tr>
<tr>
<td></td>
<td>standards along the</td>
<td>community work</td>
<td>with loss of</td>
</tr>
<tr>
<td></td>
<td>production chain.</td>
<td>but are not required.</td>
<td>interoperability,</td>
</tr>
<tr>
<td></td>
<td>However, this setting</td>
<td>However, interoperability</td>
<td>comparability and</td>
</tr>
<tr>
<td></td>
<td>does not leave much</td>
<td>standards also paved</td>
<td>clear</td>
</tr>
<tr>
<td></td>
<td>room for individual</td>
<td>parts of the way towards</td>
<td>collaboration</td>
</tr>
<tr>
<td></td>
<td>adaptation.</td>
<td>mashup</td>
<td>facilities.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLE.</td>
<td>Responsibility</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>is clearly</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>given to the learner</td>
</tr>
</tbody>
</table>

Table 1. Comparing the three scenarios according to collaboration issues and interoperability issues
With the shift from traditional and formal authoring processes and content delivery scenarios towards community driven models and personal learning environments [2], new interoperability issues arise. In the classical scenario, interoperability issues are addressed by standards that last but not least enable the tracking of learner data and assessment results, which is of importance to many who work with it. In the community scenario, interoperability issues gain importance in order to ease cross community work.

With respect to mashup PLEs, the term interoperability mainly refers to the interplay between components of the PLE: how can different widgets interoperate, while the scenario remains technically simple and open for personal adaptations [4]? However, little is said about the interoperability between PLEs of different users. How to exchange or compare results? How to measure quality, effectiveness or efficiency of highly individualized PLEs? These issues are open to further research.

Acknowledgments. This work is funded by the eContentPlus Programme of the European Commission through the ICOPER project.

References

Generalized Model for Interoperability of Data Based on Model Driven Architecture

Zuzana Bizonova
Telecom SudParis, Evry, France

Abstract. In the recent years, e-learning gained popularity among educational institutions as well as enterprises. As the result of that many commercial or open-source Learning Management Systems (LMS) were developed to manage online courses. While the usage of these systems gains recognition and acceptance amongst institutions, there are new problems arising that need to be solved. Because of multiplicity of platforms and approaches used for various systems implementation, it becomes increasingly difficult to exchange pieces of information among those systems. Applications and their data become isolated what is a clear economical concern for the future of these technologies. The present study describes a method, based on Model Driven Architecture (MDA), for integrating approaches of candidate LMS systems into a generalized architectural framework. The framework makes use of standards for description of data and metadata like learning materials (IEEE LOM, IEEE PAPI), student information (IMS LIP) or learning design (IMS LD). This platform-independent framework can be used for an automatic migration of data between various e-learning platforms.

Keywords: Interoperability, Model Driven Architecture, Learning Management Systems, Generalized Model.

1 Introduction

The sudden popularity of e-learning led to the development of a significant number of Learning Management Systems, either commercial or open source. Because of multiplicity of platforms and approaches used for various systems implementations, it has become increasingly difficult to manage interoperability of their data.

Their variety and growing number has become a true barrier for re-use of existing learning material. Creation of valuable interactive multimedia material requires a large commitment of time and resources. Due to the high costs associated with learning material development, a clear economic concern arises for the future of these technologies if the learning material and other kinds of data from LMS, such as student results and records, remain isolated with LMS applications.

Creation of valuable interactive multimedia material is demanding for time and ideas. Because of the cost of learning material development, if learning material stays isolated in applications, a clear economical concern arises for the future of these technologies. Not forgetting that other kinds of data from LMS, like student results and records, become isolated too.
The interoperability of e-learning systems has been intensively researched in recent years and several new standards have been created – for example SCORM [1]. It is a collection of standards and specifications adapted from multiple sources to provide a comprehensive suite of e-learning capabilities that enable interoperability, accessibility and reusability of web-based learning content. Other examples of standards used in LMS systems are the IMS QTI [2] standard for tests and IMS LIP [3], for encapsulating learners’ information and results.

However, most LMS systems have been created without regard to standards and therefore cannot be considered as part of an overall solution. But is it possible to achieve the goal of interoperability and data exchange even among LMS systems that are not based on standards?

The most serious obstacle to achieving this goal is that the various LMS systems have different architectures. Possible solutions would recognize these differences and try to find commonalities, and procedures to build bridges among the systems. This study aims to overcome this inherent difficulty with the current Learning Management Systems. In particular the issue of LMS interoperability among completely different architectures will be thoroughly examined.

For that purpose, we will define a new approach, based on the Model Driven Architecture [4] and using a detailed architectural analysis of candidate LMS systems which will produce different models of these systems. These models, of different levels of abstraction, will in turn be searched for commonalities and differences so that to identify unifying elements in their functionalities.

This new approach has resulted in a “three step method”. This method defines a generalized model of a LMS, as well as mapping rules that will help to translate data from a LMS system to a generalized model and again to another LMS system.

The obtained generalized model should be based on standards and other generalizing ideas so that any other LMS can be added later to this framework.

In this article the theoretical concepts of our approach will be explained that will further be used to create a generalized model of LMS systems and mapping rules between candidate systems and the generalized system.

2 Method for the Creation of the Generalized Model

Our goal is to define a generalized model of LMS system consisting of features of other LMS systems that can be mapped into it. In this part we will introduce a three steps strategy to build up the final model. This model will then represent the foundation for the data interchange among systems.

As preamble to this three step method, an exhaustive functional analysis of candidate LMS systems has to be performed. This analysis builds the sum of all functionalities found in all candidate LMS systems, its outcome is the so-called general functionality list (GFL).

The three step method will now loop through all functionalities of the GFL to perform its tasks on each of them in turn. Let us suppose that a certain functionality F is now analyzed.
2.1 First Step

The goal of the first step is to search among standards at hand to select a suitable standardized functionality SF supporting the functionality F.

It is furthermore necessary to select the most general standard from the standards relevant for the chosen functionality F. The standards are used in this concept to enhance the resiliency of our analysis with respect to changes in the field of e-learning technology. Standards are generally supported and also based on the experience of many users. Furthermore we expect that if a change happens in the future of LMS systems, this change will be reflected at the standardization level; this change can then be back-ported to our model.

The standard function SF selected to support the functionality F should be so general that it epitomizes F in as much LMS systems as possible, while being kept as specific as possible:

The mapping between F and the supporting SF can be expressed as follows:

- *Strong* support - In this case SF includes strictly F, i.e. all of F is supported by SF.
  \[ \forall F_i \in \text{LMS}_i, \exists SF \in S_i \text{ so that } SF \supseteq F_i \]

- *Weak* support - In this case SF supports only partly F:
  \[ \forall F_i \in \text{LMS}_i, \exists SF \in S_i \text{ so that } SF \cap F_i \neq \emptyset \]

- *No* support - In this case step one fails, when it cannot find any SF to support F, so that F needs to be applied "as is" to step two.

In any case the heuristic to select SF strives to minimize SF:

\[ \max(\text{SF} \cap F) \text{ while } \min(\text{SF}) \]

2.2 Second Step (Reversed MDA paradigm)

Examining now from a top-down perspective the SF obtained in step 1, we may consider that each F of the general functionality list GFL corresponds to a particular flavour or realization of the standard functionality SF, as provided by the studied LMSs.

In the second step we are creating or enhancing the generalized model by continual integration of the functionality realizations as provided by candidate LMS systems (in this study Moodle [6], OLAT [7] and Claroline [8]).

This step is performed by examining the weak support cases of step one from the point of view of missed functionality. We consider in particular the cases where a missed functionality appears in at least two LMS candidates. If such a situation appears, we tag this functionality as important (i.e. a functionality shared by LMSs, but not covered by any standards) and select it to be integrated in the model:
\[ \exists F_1 \in \text{LMS}_1, \exists F_2 \in \text{LMS}_2, \exists! cf \text{ so that:} \]
\[ cf \cap SF = 0 \text{ and} \]
\[ cf \subseteq F_1 \text{ and} \]
\[ cf \subseteq F_2 \]

where \( cf \) is a common functionality.

**Fig. 1.** Function \( F \) in LMS 1 is the red circle \( F_1 \), the same function in LMS 2 is the blue circle. Their intersection is pink in the area of standard. The important missed functionality is the dark pink intersection of \( F_1 \) and \( F_2 \) – not covered by the standard but still in the intersection of definitions of functionalities of more than one LMS system.

For this purpose we use the Model Driven Architecture (MDA) that was previously described. Just to remind the reader of the concept, MDA is a way to organize and manage system architectures. The building of the system can be organized around a set of models by imposing series of transformations between them. The whole system creates an architectural framework of layers and transformations.

OMG defines three levels of abstraction (fig. 2) [12]:

- **Platform Independent Model (PIM)** – this model provides adequate functionalities, structure and behaviour of the system,
- **Platform Specific Model (PSM)** – combines PIM with specific detail concerning the way in which the system uses a certain platform – it can be automatically transformed into the implementation code.
- **Implementation**

**Fig. 2.** MDA Concept
MDA principles are the background for the solution of the proposed problem with the LMS system integration.

**Reversed MDA Paradigm**

One of the reasons why to use MDA was that analyzing the system on different levels of abstraction helps us to understand the system better. The abstraction of the candidate LMS systems to the platform independent level can give us the necessary “look from above” to see the commonalities and differences among various systems. What we have at first (fig. 3) are the implementations of the LMS systems in frameworks of various technologies. These implementations can be abstracted to the PSM models and further to PIM models where technology used is irrelevant. At this level finally we can see the commonalities between various systems. Now that we got rid of implementation details of each LMS system, we are able to clearly see for example F1 and F2, it means how a certain functionality is realized in the architectures of various systems. We can describe and analyze them. The outcome of this step is a set of commonalities that can be further used to create (in case of no support from the first step) or enhance (in case of weak support) the General PIM by important missed functionalities.

This analysis also gives the foundation for the mapping rules from the functionality F1 in the LMS1 to the functionality F in the General PIM and from the functionality F in the General PIM to the F2 functionality in the LMS2.

![Fig. 3. Reversed MDA concept for creation of General PIM.](image)

The previously described step repeats for each separate functionality in the system. Figure 4 shows the integration strategy to create the General PIM. For a certain functionality, PIM models of candidate systems are enhancing the General PIM until the model is saturated. Then we continue to enhance the generalized model with another functionality.

The systems used in our research were selected based on their variety of architecture, structure and technology used. It is to ensure that the General PIM contains a large combination of various functionality realizations and therefore many other systems can afterwards use this model.
2.3 Third Step

The goal of the third step is to create mapping rules of the PIM models of candidate systems to the General PIM, and vice versa. Practically it means to create translation tables for data structures of a LMS system to the generalized system and the other way round.

Again, referring to the fig. 1 we can see that it is considerably easy to map functionality F1 of LMS 1 to the General PIM because General PIM should cover most of the F1 functionality (the intersection F1 ∩ SF plus F1 ∩ F2) and this part can be just “translated” to the terms of General PIM. The question is what to do with those values that are not covered by General PIM.

As we mentioned before, these missing features appear only in one of the candidate systems, therefore they are not incorporated in the General PIM. It means that we need to transfer them only in those cases when we transfer data from a system to the same system and we need to save also these extraordinary data. In this case we can use the class G_extra_metadata that will be introduced later in the sixth chapter. This class serves as a list of any definable attributes and their values that can be added to the main class G_repository_entry of any item in the model. This way we are able to create mapping rules for both the data that are incorporated in the General PIM as well for those that are not.

Fig. 5. Mapping rules – from Moodle to the General PIM, from General PIM to OLAT, and the other way round.
Systematic mapping between the candidate systems data and the data of the generalized system requires definition of mapping relations between entities of the target systems. Such relations, or mapping rules can have forms of 1:1 (direct mapping), 1:n (divergent mapping) or n:1 (convergent mapping). The set of mapping rules is constructed both ways: describing the transformations from the candidate system to the generalized model and the other way round. The direct mapping case is trivial. We simply “translate” one attribute to another one. The table will look like this:

<table>
<thead>
<tr>
<th>General PIM</th>
<th>Candidate PIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>G_class.visible</td>
<td>Candidate_table.visible</td>
</tr>
</tbody>
</table>

Tab. 1. Relationship 1:1.

In the case where there is a convergent or divergent mapping, the situation has to be analyzed in more detail.

- simple convergent/divergent mapping
  
  For example there can be a redundancy case where a value is mentioned twice in our generalized model or in the candidate model. Let us say there is just one type of name in the candidate table, but two types in the generalized model. The table can look like this then:

<table>
<thead>
<tr>
<th>General PIM</th>
<th>Candidate PIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>G_class.shortname</td>
<td>Candidate_table.name</td>
</tr>
<tr>
<td>G_class.longname</td>
<td>Candidate_table.name</td>
</tr>
</tbody>
</table>

Tab. 2. Relationship n:1.

On the other hand 1:n relationship can be written in this manner:

<table>
<thead>
<tr>
<th>Candidate PIM</th>
<th>General PIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candidate_table.name</td>
<td>G_class.shortname/ G_class.longname</td>
</tr>
</tbody>
</table>

Tab. 3. Relationship 1:n.

We do not tackle the situation of n:m relationship because of its great complexity. Such a relationship is however always composed of n:1 and 1:m relationships that can be implemented.

- complex convergent/divergent mapping

There can also be a complex value consisting of many objects that need to be combined. The actual combination law has to be defined manually (numbers: addition, strings: concatenation in simple cases). See an example on the table.
### 3 Conclusion

The General PIM and the mapping rules represent the architectural framework that enables data interchange among LMS systems. Any piece of data can be translated to the General PIM with the mapping rules and translated again to the same piece of data in another system. As soon as any module is added into the General PIM and mapping rules of the LMS systems are written for it, we have gained the framework for the data transfer.

This generic approach can be used to enhance interoperability among systems that have not been created based on any standard as such case is not rare in the current technology enhanced learning environment. Such approach has been further developed and applied on three candidate systems in the dissertation thesis that was defended in September 2008 and the theoretical concepts have been proved by a demonstrator that successfully transformed various kinds of data between candidate systems, like tests and test questions, student results or even forums, chats and assignments [14].

### References

2. IMS QTI, (http://www.imsglobal.org/question/)
4. MDA, (http://www.omg.org/mda/)
8. Claroline project – website (Nov 2007), (http://www.claroline.net/)
(http://www.etwinning.sk/2008/Middle).
Differentiating and Defining Portlets and Widgets:  
A Survey Approach

Effie L-C Law\(^a\), Daniel Müller\(^b\), Anh Vu Nguyen-Ngoc\(^a\)

\(^a\)Department of Computer Science, University of Leicester, LE1 7RH Leicester, UK  
\(^b\)IMC information multimedia communication AG, 69115 Saarbruecken, Germany

\(\text{*elaw|anhvu}@mcs.le.ac.uk, \text{Daniel.Mueller@im-c.de}\)

Abstract. Widget is not a new computing concept, but it has recently attracted much research interest because of its potential uses in a range of technological products/services. In assuming the challenge to develop an evaluation scheme for widgets/widget-based applications, some conceptual ambiguities about widget, especially its distinction from portlet, need to be clarified first. A survey aiming to collect opinions in this regard was conducted. Qualitative data were analyzed with Cmap (a content map tool) and Nvivo8 (a content analysis tool). Widgets invite more diverse interpretations than portlets, for which specifications have existed for some time. Relationships between these two entities are not consistently understood by respondents. A consensual definition of widget entails more scientific discourses in the future and will have significant implications for developing a robust evaluation framework for widget-based platforms.

Keywords: Portlet, Widget, Concept map, Content analysis, Open and responsive learning environment

1 Introduction

Widget is not a new concept in the computing world. Some researchers claim that the emergence of widgets (or their precursors) can be traced back more than 25 years ago when GUI was first designed for home use [1]. Since 2003 when desktop widgets were brought to Mac OS X users, widget has become a buzzword and prevailed in various contexts such as websites (especially social network pages), mobile devices, and desktops. In the same year, the first Java Portlets Specification JSR 168 (http://jcp.org/en/jsr/detail?id=168) was published; it addresses the issue of interoperability of portlets across different portal platforms. In contrast, it was not until very recently in July 2009 when W3C released Widgets 1.0 Specification\(^1\). Despite their different paths of evolution, the similar characteristics and functions of

\(^1\) Accordingly, widget is defined as: "A widget is an interactive single purpose application for displaying and/or updating local data or data on the Web, packaged in a way to allow a single download and installation on a user's machine or mobile device. A widget may run as a stand-alone application (meaning it can run outside of a Web browser), and it is envisioned that the kind of widgets being standardized by this effort will one day be embedded into Web documents."
portlets and widgets have tempted people to use the two terms synonymously. Interestingly, each of them is associated with a bunch of phrases or names that may (or not) imply their key properties, such as “reusable Web module”, “pluggable UI components” for portlets and “gadgets”, “snippets”, “flakes” for widgets. Out there in the scientific and grey literature, there exist a number of formal as well as quasi definitions for portlets and widgets. Due to their crossroad developments, researchers and practitioners are somewhat confused how widgets should be distinct from portlets or not at all. Whilst several attempts to develop and validate an evaluation scheme for portlets and portals have been undertaken (e.g. [2]), to our best knowledge, it is yet to be done for widgets and widget-based applications. We intend to assume this challenge. The first task we need to tackle is differentiating portlets from widgets, i.e., to what extent they are similar or different? Answers to this key question will lay the cornerstones for our further work, because they will have the significant implication whether the existing evaluation frameworks for portlets can somehow be adopted and adapted for widgets. Specifically, widgets, given their versatility and flexibility, will presumably play an increasingly important role in technology-enhanced, open and responsive learning environments (cf. http://www.role-project.eu).

As an effective and efficient means to collect views from experts working on the related topics, we have developed a survey (Section 2). The data enable us to clarify certain conceptual ambiguities and provide us a solid foundation to proceed with our plan to develop a valid evaluation framework for widgets/widget-based applications.

2 Methodology

Survey Design and Administration: To maximise the response rate, our survey is designed to be succinct and precise. It consists of four main open-ended questions, which are further divided into sub-questions (Table 1). The survey has been administered via personalized emails to a group of European experts on portlet- and widget-related topics. It has also been posted onto a personal blog (http://www.pontydysgu.org/2009/06/portlets-and-widgets/) and LinkedIn ROLE community (http://www.linkedin.com/groups?gid=1590487). We have gathered 24 completed responses with various levels of details. They are designated as R1, R2, and so forth. Most of the respondents are developers with strong technical background whereas the other respondents are more mediators (who collect technical requirements and communicate them to the development team) or active users of portlets/widgets. Four of the respondents are female.

Data Analysis Tools: To enhance our understanding of survey data, we employ several data analysis tools, viz. CmapTools and Nvivo, to consolidate our empirical findings. Here we briefly describe each of them:

CmapTools is a free software application enabling users to construct, navigate, share and criticize knowledge models represented graphically as concept maps. Concepts are enclosed in circles or boxes. Related concepts are linked hierarchically by connecting lines annotated with words/phrases to specify their relationships. A concept is defined as a perceived regularity in (records of) objects/events designated
by a label [3]. With CmapTools we have built concept maps on the responses to Q.1 and Q.2 (Table 1).

Table 1: Survey questions

<table>
<thead>
<tr>
<th>Q1. Questions about Portlets</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a) Please give your <strong>definition</strong> of portlets</td>
</tr>
<tr>
<td>1b) Please list specific <strong>characteristics</strong> (=attributes, properties) of portlets</td>
</tr>
<tr>
<td>1c) Please list specific <strong>features</strong> (=functionalities) of portlets</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Q2. Questions about Widgets</th>
</tr>
</thead>
<tbody>
<tr>
<td>2a) Please give your definition of widgets</td>
</tr>
<tr>
<td>2b) Please list specific <strong>characteristics</strong> (=attributes, properties) of widgets</td>
</tr>
<tr>
<td>2c) Please list specific <strong>features</strong> (=functionalities) of widgets</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Q3. Please tell us, what do YOU consider as the <strong>major differentiator(s)</strong> between:</th>
</tr>
</thead>
<tbody>
<tr>
<td>3a) Portlets and Widgets? (cf. Wikipedia on Web widget)</td>
</tr>
<tr>
<td>3b) Widgets and Java Applets? (cf. W3C Widget requirements)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Q4. Please share with us YOUR ideas how to evaluate:</th>
</tr>
</thead>
<tbody>
<tr>
<td>4a) Portlets? 4b) Widgets?</td>
</tr>
</tbody>
</table>

Nvivo 8 is a proprietary software package supporting users to analyse qualitative data of different sizes, ranging from simple text to multimedia data. It facilitates importing, classifying, sorting, and some other manual qualitative data analysis tasks. It also provides the possibility to query data and visualise them with charts and models. We have deployed Nvivo 8 to code the survey responses. The coding results enable us to further explore the data in a greater detail, both manually and with the use of Nvivo 8, to identify the concepts commonly elicited from the respondents and to highlight their possible misunderstandings about and contradictory interpretations of portlets and widgets.

3 Concepts Associated with Portlet Definitions and Features

In response to Q.1 of the survey, a variety of concepts are elicited. Whilst most respondents tended to provide a succinct statement for defining portlet (1a), a few of them provided an elaborated answer. Besides, quite a number of respondents remark that it was hard to differentiate between characteristics and features and thus collapsed their answers to 1b and 1c into one. Hence, we compiled the answers to the three parts of Q.1 and constructed concept maps accordingly.

Fig. 1 shows that the main concepts associated with portlets are: portal, web-based application, and widget. Each of them is connected to some sub-concepts. Presumably the diagram is self-explanatory. Note that what depicted are respondents’ interpretations of portlets, which may or may not be valid.
Fig. 1. Concept map of a comprehensive portlet definition
To complement as well as supplement the above findings, Nvivo results allow us to explore individual ideas more in detail, how often they are claimed and by whom, and how they are connected.

As portlets are standardised with JSR 168/286 developed under the Java Community Process, it is logical that eight of the respondents claim that portlets are Java-based and another four mention that portlets are independent, small applications. Eleven out of 24 respondents define a portlet as a user interface component. 62.5% of the respondents state that portlets must be associated with or run in a web portal. Besides, four respondents remark that portlets should serve as a functional part of an application. Interestingly, several respondents explain portlets in terms of widgets: a portlet is a “widget to be integrated in a targeted Web portal”, “a more comprehensive widget”, or “not far from widget”.

In describing characteristics and features of portlets, there are communal as well as contradictory responses. Seven respondents claim that portlets are embeddable or pluggable. Another four remark that portlets “produce document fragments” that can be combined in Web pages. In addition, the functionality of portlets depends on the portal in which they are running, as clearly defined in the JSR: “The content generated by a portlet is also called a fragment ... a piece of markup (e.g. HTML, XHTML, WML).... The content of a portlet is normally aggregated with the content of other portlets to form the portal page”. Some respondents assert that portlets can be configured and customisable, though the configuration can be restricted.

Interoperability of portlets is another key feature mentioned by ten respondents. Some of them write that a portlet can “interchange with or possibly talk to other portlets”, and portlets can “communicate to a certain extent (difficult problem, though)”. Some refer to standards and specifications that support the portlet interoperability, such as “event-based inter-portlet communication (IPC)” and “services-API to other portlets”.

Despite the relatively established status of portlets, respondents hold contradictory views about their characteristics and features. One respondent claims that a portlet is a client-sided component whereas two reply that it is server-side technology. Some mention that a portlet “highly relies on the application logics on the server”. Others say that a portlet has “interaction with backend services” or follows “client/server communication” and “MVC paradigm”, in which the application logics are carried out mostly at the server side and the client side is used mostly for displaying the data process results. Other conflicting views include static vs. dynamic content generation and lightweight vs. heavyweight.

4 Concepts Associated with Widget Definitions and Features

As explained in Section 3, we merged the responses to the sub-questions when performing data analysis. Fig 2 illustrates the Concept maps on the concepts elicited from the respondents when interpreting widgets. The diagram is intuitive as well as informative.
Fig. 2. Concepts of a comprehensive widget definition
Apparently, there is no clear definition about widgets. The respondents provided a diversity of concepts in connection to widgets. One respondent admits that “widgets are less well and less explicitly defined than portlets”. Twelve (or 50%) respondents remark that widgets are simple and small applications. Four respondents mention that widgets can be defined user interface components (cf. 11 mention so for portlets). According to three respondents, ‘desktop widgets’ should be explicitly distinguished from ‘web widgets’. As pointed out by six respondents, developers tend to use a lot of HTML, Java and Javascript to develop widgets; however, they took only web widgets into account. There are some possibly controversial and contradictory concepts in the widget definitions. Here are some of the interesting findings:

- **Platform dependencies**: Eight respondents only refer to widgets as web applications or as applications that run only on web platforms. Most other respondents tend to think that widgets can run on different platforms, including web, desktop and mobile devices;
- **Implementation**: According to one respondent, widgets are “simple HTML and Javascript applications”. Similarly, another asserts that “widgets are small applications built using Javascript, CSS and HTML”. There are also other responses that tend to support those statements, including “widgets tend to use web standards (HTML, CSS, Javascript)” or “html snippets plus javascript code”, “mainly consisting of Javascript and XML”, “using a lot of Javascript”.
- **Functionalities**: One respondent mentions that widgets are “not integrated functionally into an application, but only visually”. In fact, a widget can also perform rather complicated functionalities. Many respondents agree that a widget can be seen as an entity providing specific and useful functionality. They remark that a widget can “have any functionality, except heavyweight application functions”, can provide “quite complex application logics at the client (in the browser) in combination (or not) with functionality from a server”, can be “more advanced in providing functionality” or can “process data as well as display data and result of processed data”. Besides, one respondent argues that widgets “are about the web-based visualisation a functional part of a web application”. Similarly, another respondent states that widgets are “developed for being included and displayed in a software application”. However, many respondents refer to a widget as an application itself.

Concerning the characteristics and features of widgets, we identify both consensual and contradictory concepts in the survey responses. Many respondents agree that widgets are possibly configurable and/or customisable. However, one respondent argue that a widget is “customisable for developers, for users it is just an ‘interaction element’ or ‘graphical unit’...” and another remarks that only “widgets layout format is configurable by users”. Various properties of widgets are mentioned by respondents: light-weight, single-purpose, independent from the framework or infrastructure in which they run, interactive, reusable, portable or embedded. Concerning the technologies used for widgets development, beside the discussions on the definitions, five respondents explicitly mention AJAX (i.e. asynchronous JavaScript and XML).

The issue of widget interoperability is highly relevant as it can help develop useful and powerful mashup applications such as personal learning environments (PLEs). Some respondents tend to support that widgets are interoperable, but some
others are rather against it. The latter argue that technically it is very difficult to support widget interoperability because standards for inter-widget communication is still lacking.

5 Comparisons between Portlets and Widgets

In discussing the difference between a portlet and a widget (Q.3 in Table 1), the respondents again expressed diverse opinions. In fact, many of them reiterated the concepts that they had already provided for Q.1 and Q.2. An answer with a slightly high level of agreement is on the platform where portlets and widgets can run: Portlets need to run in a portal while the widgets can run in different platforms. The context in which widgets can be used is broader (i.e., widgets are platform independent). Another point is that widgets seem to use more client-side technologies and more lightweight. In contrast, portlets are more complex and heavyweight. Apparently, respondents hold different views how portlets are distinct from widgets in terms of ease of implementation and integration (widgets easier) and pace of evolution (widgets faster). An interesting argument is the subsumptive relationship between the two objects with some respondents seeing portlets as a subset of widgets.

To sum up, we tend to agree with the expressed view that “widgets are less well and less explicitly defined than portlets”. As portlets are established technologies standardised in the Java community, developing portlet-based applications seem to be clear and straightforward though it may be complicated in the sense that developers should follow some rules defined in the specifications. However, the portlet usage is quite limited in the web portal. We also agree with some respondents that portlets are subsumed by widgets. Widgets can be used as standalone applications or combined with other widgets to form bigger applications, especially when the future widget-widget communication APIs are standardised. Widgets do have a great potential.

6 Concluding Remarks

While the concept of widgets is not new, how they can be exploited by today’s technologies is perceived to an innovative challenge. Resolutions to this challenge will presumably have strong impacts on the future development of a range of products/services, given (potential) versatile functionalities and platform independence of widgets. The hype surrounding widgets has recently been turned into something that takes some time to ripe - the latest W3C working draft on Widgets 1.0 API and Events which is directly related to the widget development. Meanwhile, in the market many companies offer widget development using different technologies and programming languages. It is deemed important to enable widgets to be deployed in various contexts and platforms.

Another challenge related to widget development is widget evaluation (i.e. design and evaluation of two faces of the same coin). As mentioned earlier, what has driven us to undertake the task of distinguishing portlets from widgets conceptually is a more ambitious goal to develop an evaluation scheme for widgets/widget-based
applications. This is where Q.4 of our survey comes into play. We will tackle this challenge as our next contribution to the wider research community.

Some reflections on the usability and usefulness of the three data analysis tools deployed for this study are presented here. While CmapTools is powerful to visualize results in an intuitive manner, it is not easy to deploy as it entails quite a lot of manual work. Similarly, Nvivo 8 allows users to abstract relationships among concepts (i.e. nodes) at different levels and support other qualitative data management functions. However, it is a heavy application in the sense it necessitates some form of training to use it effectively and non-trivial manual operations.

References