FROM SILOS TO MASHUPS

CONFORMANCE THROUGH WIDGET-BASED MASHUP APPROACHES TO LEARNING ENVIRONMENTS

Fredrik Paulsson, Umeå University, Umeå, Sweden, fredrik.paulsson@umu.se

Abstract

Different types of digital learning environments have evolved over the years and there is an ongoing evolvement of different concepts and technology. During the last 10-15 years Learning Management Systems have dominated. In recent years there has been a development of concepts like Personal Learning Environments using mashup technology. This paper presents a study of a mashup approach to the digital learning environment using web widgets. A prototype was developed and tested, covering some technological aspects such as modularity, integration and adaptability as well as pedagogical aspects, such as pedagogical flexibility, technological responsiveness and pedagogues roles and attitudes. The study shows that widgets and mashup technology can be used to construct digital learning environments that have the potential to replace traditional LMSes and that pedagogues are positive to the adaptive learning environments and flexibility that they represent.

Keywords: LMS, MUPPLE, web widgets, mashup, VLE, PLE, e-learning

1 Introduction

Different types of *Digital Learning Environments (DLE)* have evolved over the years and there is a strong momentum and an ongoing progression of different ideas and concepts for the DLE. During the last 15 years a lot has revolved around the idea of Learning Platforms, such as *Learning Management Systems (LMS)*. These systems are commonly set forth as a common solution for a range of educational needs – much like a "Business System" for learning and education. However, LMSes have been criticized for being inflexible and hard to adapt to different pedagogical contexts and needs (see, e.g., (Paulsson, 2008), (Paulsson & Naeve, 2006b) and (Atwell, 2007)). LMSes are also commonly criticized for being too much about the administrative aspects of learning and education and only little about pedagogical activities and pedagogical processes. Hence, having a strong focus on *Learning Management* rather than on the actual learning activities and pedagogical processes per se - as the term also suggests. From a system perspective LMSes are often criticized for being designed and implemented in a silo-like fashion, contributing to lock-in effects of information and processes, preserving old-fashioned system borders. The criticism is actually similar to the critique often heard about business systems in general. There is also a built-in conflict between the development and implementation of systems like LMSes on the one hand and the development and implementation of social software and Web 2.0 on the other hand. While many LMSes currently in use try to create a well-defined kind of "shielded community" for learning, web 2.0 is associated with open communities, global social interaction and open information services that can be used as building blocks for new services[1] - such as for a *Personal Learning Environment* (PLE), as described in (Atwell, 2007), (Jones, 2008) and (Wilson et al., 2006).

[1] Observe that the notion of services for Web 2.0 refer to services that target users and are not equivalent to services as in *Service Oriented Architectures (SOA)*, which is a software design paradigm (Erl, 2007). While the technology platform

In order for such web services to be used as building blocks in mashups (i.e. a composition of services) the building blocks need to be well defined and with well-defined interfaces. Many web 2.0 services use proprietary interfaces such as the Twitter API, the Facebook API or APIs from Google and/or they use lightweight interfaces and protocols, such as RSS or Atom. This works well in many cases, but in order to build more sophisticated mashups there is a need for more standardized and sophisticated interfaces and concepts for interaction that can be generalized as exemplified in (Paulsson & Berglund, 2008), (Paulsson, 2008), (Wilson, Blinco, & Rehak, 2004) and (Mulligan, 2009). This can obviously be accomplished by using advanced proprietary APIs, as illustrated in (Paulsson & Berglund, 2006), but from a wider perspective, common open standards are preferable, and in the long run a necessity for mashed-up learning environments to be practical. This is also one of the issues the study discussed in this paper was set out to examine, addressing the question: "What are the possibilities for constructing mashed up learning environments based on available open standards?" The study also briefly evaluates and the relationship between mashup approaches and adaptability/responsiveness to pedagogical requirements.

The next section briefly describes the state of art, followed by a discussion on some central concepts and ideas, followed by a presentation of the study and the prototype implementation of a mashup learning environment. Finally the results of the study are discussed in the light of the ongoing progress and previous research in the field.

1.1.1 LMSes and PLEs

Simply put, a *Personal Learning Environment (PLE)* can be described as a learning environment where the learner is in focus as well as in control of the learning environment. However, the main objective of the PLE is to put the learner in control of his own learning rather than in control of the learning environment, even though these two are obviously related. Learning is regarded as a constant, ongoing process, as is the evolvement and change of the learning environment. The learning environment needs to be responsive and adapted to different contexts, needs and pedagogical requirements. There is also a focus on social interaction and the use of Web 2.o technologies in general. These are qualities that are commonly emphasized, such as in (Paulsson, 2008), (Atwell, 2007), (Wilson et al., 2006) and)(Jones, 2008

While LMS-like system are typically implemented by most educational institutions, the movement within the teaching community as well in the research community is towards more adaptive and responsive learning environments, similar to PLEs, see e.g., (Atwell, 2007) (Wilson et al., 2006) (Jones, 2008) (Severance, Hanss, & Hardin, 2010) and (Palmér, Sire, Bogdanov, Gillet, & Wild, 2009). However, while pedagogical concepts like responsive learning environments are attractive, the technology currently in use doesn't support it very well. At the same time, education need specialized services for dealing with pedagogical requirements, such as Personal Development Plans (PDP), digital portfolio, services for discovery and integration of digital learning resources, and so forth, which are resulting in several good and useful tools for learning, but they are not well integrated with the rest of the VLE (Paulsson, 2008) (Sultan, 2010) (Severance et al., 2010). These and similar issues are often addressed through different approaches to system integration on a per system basis, such as by using proprietary APIs or more general integration by Web Service technology, such as described in (Paulsson, 2008) (Erl, 2007) (Paulsson, 2006) (Ogrinz, 2009) and (Gonzalez, Penalvo, Guerrero, & Forment, 2009). However, such approaches to building the learning infrastructure has turned out to be problematic for several reasons. Firstly, it becomes expensive to integrate "per system", using proprietary APIs. API integration tends to make the systems hard coupled, which supresses flexibility, as described in (Paulsson, 2008) and (Brereton & Budgen, 2000). Secondly, using (commonly SOAP-

---

underlying Web 2.0 services may very well be a SOA platform, there is an unfortunate mix-up of those two rather different notions of services when discussing Web 2.0.

based) Web Service technology tend to become very complex as well as expensive, adding an cost, as well as technical, overhead (Paulsson, 2008) (Brereton & Budgen, 2000) and (Preciado, Comai, & Sánchez-Figueroa, 2005) and not really suitable for the kind of integration needed for adaptive and responsive learning environments controlled by the user. And thirdly, by mixing a monolithic concept, like the LMS with a modular service based approach some of the technical flexibility needed for dealing with some of the pedagogical requirements is lost, as illustrated in (Paulsson, 2008) (Atwell, 2007) (Wilson et al., 2006) and (Palmér et al., 2009).

1.1.2 Widgets and Mashups

In recent years there has been a general development on the Internet towards modularity and an alternative kind of loosely couple services driven by less complex and more web friendly service integration, such as using RESTFul APIs [21] and lightweight APIs and protocols, such RSS and Atom combined with widget and mashup technologies, see e.g. (Yu, Benatallah, Casati, & Daniel, 2008) (Wong & Hong, 2008) and (Hoyer & Fischer, 2008).

In conjunction with this development, a number of standards and specification, dealing especially with issues related to widgets and mashups, have been developed. There are several (but similar) definitions of a mashup. A web-mashup is commonly defined as being a combination of different services on the web in a way that create a new composite application (or service) with an added value. A widget-based mashup obviously uses widget technology and is typically constructed using a mashup environment, such as Netwibes or iGoogle, see e.g. (Casquero, Portillo, Ovelar, Romo, & Benito, 2008), (Godwin-Jones, 2009) and (Hoyer & Fischer, 2008). A mashup can also be created by very simple means, using simple web tools that allow users to combine services on the web by matching and mixing information using lightweight interfaces such as RSS or Atom. However, in such cases it is mainly about mashing up information and not that much about mashing up functionality and services in a way that goes beyond the delivery and consumption of information – even though information mashups can be valuable in many cases as part of a digital learning environment irrespective of type.

Even so, if you are a developer or an experienced user you might want to use one of the more sophisticated approaches that are available for the development of web-based applications, or RIA (Rich Internet Applications) (see (Preciado et al., 2005) and (Fraternali, Rossi, & Sánchez-Figueroa, 2010)), as it is sometimes referred to, that has the potential of moving computing from the desktop to the web.

Standards are a necessity to make Widget-Mashups work in a wider perspective and in symbiosis with the development Widget and Mashup technology, a number of standards and specifications has evolved and most distinguished among those are most likely the Open Social specifications, described in (Mitchell-Wong, Kowalczyk, Roshelova, Joy, & Tsai, 2007) and (Häsel, 2011), and the package from W3C, described in (Wilson, Sharples, & Griffiths, 2008) and (Wilson, Daniel, Jugel, & Soi, 2012).

The Widget and Mashup technology concept stands out as exceedingly suitable for the next generation of learning environments, fulfilling the flexibility requirements for personal and responsive learning environments by providing standardized frameworks for modularity and loose integration on the web. As discussed in (Paulsson, 2008), (Wheeler, 2009), (Verpoorten, Westera, & Specht, 2011) and (Severance et al., 2010), there is a relation between the flexibility provided by modularity and the pedagogical flexibility needed to make the learning environment adaptive and responsive. Hence, using modular approaches for the design and implementation of digital learning environments can address some of the LMS related issues that were described in the previous section.

The terms Widgets and Mashup will from now on referring to W3C or OpenSocial widgets and Widget Mashups in the context of this paper.

The study described in this paper was carried out in a project called WiLearn. Before describing the WiLearn project some brief background will be given in the next section.

1.1.3 Background and context

The WiLearn project is the successor of two previous research projects dealing with modular digital learning environments and the relation between modularity, technical flexibility and pedagogical adaptability and responsiveness. The first project, called the Virtual Workspace Environment (VWE) was started in 1998. The objective of VWE was to implement a modular digital learning environment that also moved the computer desktop to the web. The underlying idea was that teachers and students should be able to construct and change their own, as well as their shared, learning environment by combining building blocks (components) called VWE Tools. The learning environment didn't exist until someone assembled it for a specific pedagogical purpose, using components suitable for supporting that specific pedagogical scenario. VWE was constructed using Java RMI on the server and Java Applets on the client (in the browser) and an important part of the VWE was a collection of server-sider services that kept the environment together by dealing with the internal communication of components, session management and other system stuff. The technical solution and architecture made VWE somewhat proprietary and there were no standard components that could be used out of the box. Everything had to be adapted for VWE. The use of Java RMI and applets created an overhead that made it to scale and it put a lot of demand on the browser, see (Paulsson & Berglund, 2003). Some of those problems were addressed by the use of SOAP web service, but even though this solved some of the problems, it also contributed to new overhead, see (Paulsson & Berglund, 2008).

Taken altogether this made it very hard to make comprehensive field studies with teachers and students. However, in spite of this VWE contributed with some valuable knowledge about the distinct relationship of technical flexibility, provided by modularity and the pedagogical flexibility and adaptability as a result. This was discussed in detail in (Paulsson, 2008). The VWE project also pointed out the importance of standards. General standards as well as Learning Technology Standards (LTSC) that are needed to keep the learning processes and the digital learning content together, something that are discussed in (Paulsson & Naeve, 2006a), (Salmenjoki, Paulsson, & Vainio, 2007), (Nilsson, 2001) and (Varlamis & Apostolakis, 2006).

WiMupple was the name of the second project and the objective of WiMupple was to compare a Widget-based Mashup solution to the Java RMI and SOAP-based approaches of the VWE. A simple prototype was developed and compared to the properties of VWE. Some new conclusions could be drawn about the importance of web friendliness and some of the findings from the VWE-project, such as the importance of standards, could be verified once again. The main differences were less technical complexity and overhead in the implementation and the ability to use standardized components (i.e. widgets) from third-party. The use of the Wookie widget server to implement the W3C widget standards made this possible; see (Wilson et al., 2008). The ability to use widgets developed by others is important as it facilitates reuse and stimulates development. Even though the Wookie server was used, some in-house development was needed in order to make the WiMupple platform usable. This resulted in a couple of REST APIs based on *Yahoo Querying Language* (YQL) (2011) and some proprietary functionality, which actually added some similar problem as VWE. However, this was necessary to develop the same level of functionality as in VWE The WiMupple project was discussed in (Paulsson, 2012).

The next part of this paper presents the WiLearn project the study carried out, followed by a discussion of the results and some ideas for future studies.

2 Objective

The objective of the WiLearn project was to study how teachers approach modular digital learning environments and to study if and how the technical flexibility offered contributes to the view of the pedagogical possibilities. Another objective of the study was to implement a prototype of a generic mashup environment for constructing digital learning environments based on widget and mashup technology, without the limitations of the VWE and the WiMupple environments. The purpose was to use the prototype for the study fulfilling the first objective.

It should be noted that the basic idea is still the same as with the previous studies: that teachers and students should be in control and that the learning environment can be assembled for a specific situation based on pedagogical needs and requirements. Different types of digital learning environments can be constructed, such as PLEs or LMS-like learning environments, depending on the type of widgets and supporting backend systems that are available and used.

3 Methodology

An experimental approach was used for the study in the WiLearn project. The approach was inspired by the design-based research methodology described in (Wang & Hannafin, 2005) and in (Anderson & Shattuck, 2012). The intension was to create a setting that was as similar to a real-life pedagogical scenario as possible by developing a functional prototype that didn't restrict the field part of the study more than necessary.

3.1 Technology settings

An important premiss was to avoid unnecessary development and avoid developing everything from scratch, which was the case in the VWE-project. There is a multitude of ongoing development and projects addressing widgets and mashups and one of the first design decisions were to go as mainstream as possible and existing work has been used whenever possible. The main platform used was the Apache Rave framework. Apache Rave is described by the development team as a *"…web and social mashup engine that aggregates and serves web widgets…"*[2]. The reason for the choice of Apache Rave was mainly that, even though Apache Rave is in an early stage of development, it is based on two solid Widget platforms providing stable widget frameworks. The main components of Apache Rave are the Apache Wookie server, serving W3C widgets, and the Apache Shindig, serving OpenSocial widgets, see (Wilson et al., 2008) and (Häsel, 2011). This also means that WiLearn has the ability to use both W3C widgets and OpenSocial widgets for mashups, which increases the number of available widgets. Another important reason for using Apache Rave is that it takes care of things like session management, management of user workspaces and login/single sign-on (SSO). Especially SSO is a central feature for mashups, as the user will actually be using several systems, even though it may have the look and feel of one integrated system. Besides, Apache Rave can support different ways of handling login, which makes it easier to integrate with existing user management and system legacy. Figure 1 illustrates the components and architecture of Apache Rave.

Using Apache Rave as the basis, the features needed to implement a simple but functional and flexible learning environment were developed. Both OpenSocial widgets and W3C widgets were used and the choice of widget framework mainly depended on two things: whether there were existing widgets that could be used and the type and character of the functionality to be implemented. It turned out that a large part of the needed widgets were already available as widgets developed by others. However, these often needed some adaption in order to be suitable for our purposes and to be suitable for use in

[2] The Apache Rave website. Retrieved from http://rave.apache.org/ 2013-03-01

a learning environment. Some of the core (administrative) features that are normally found in the LMS were implemented using widgets, but the focus of the widget implementation was on widgets adding pedagogical value. Besides the possibility to use widgets that ere adapted by for the sake of the study, a possibility to search two different “Widget stores” was implemented. In total over 300 000 widgets that could potentially be found and used to add new and/or alternative functionality to the learning environment.

The technology requirements at the client-side are very low as a consequence of going mainstream. There is no need for software installations or plug-ins. The only demand on the client is a reasonably new browser and a stable Internet connection. This is a distinct difference from both the VWE project and the WiMupple project where certain browser versions and software was a requirement.

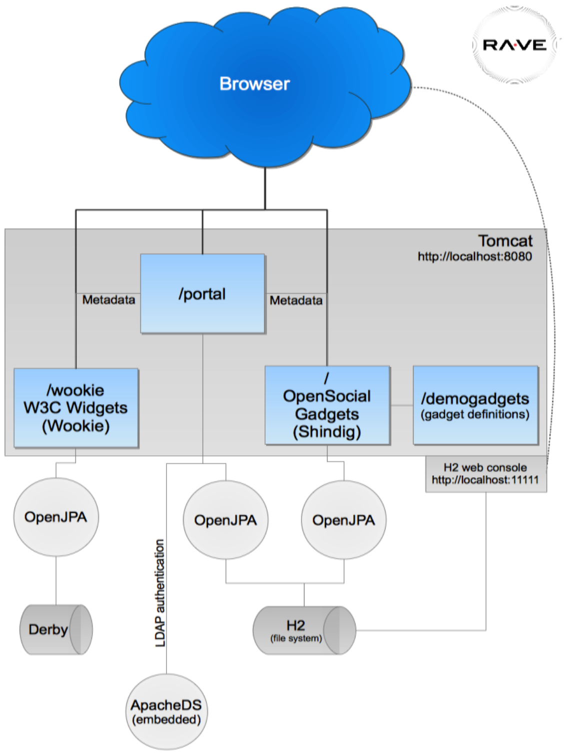

Figure 1. Components and architecture of Apache Rave[3]

[3] The Apache Rave website. Retrieved from http://rave.apache.org/ 2013-03-01

3.2 The field study

The prototype environment that was described in the previous section was used for a field study. The field study involved in total 34 pedagogues from public schools and was carried out as a number of workshops involving 6-12 pedagogues each time. Informal conversations took place during the workshops and from the perspective of the study, these were threated as a form of contextual interviews, even though the form was very informal and the main activity during the workshops was the work with the digital learning environment. The focus of the workshops was on pedagogues composing their own digital learning environment using the WiLearn platform and widgets.

In order to get the most out of the “collective intelligence” of the pedagogues participating in the field study, the pedagogical scenario and the composition of the learning environment was carried out as a collective effort, with support from a developer in order to be able to encounter for issues that may occur from the fact that the WiMupple platform is still in a prototype stage and that problems may occur from time to time.

4 Results and discussion

First of all, it is obvious that a lot has happened technology wise since the first attempt towards a modular digital learning environment that was made in the VWE project. The fact that there is now a common concept a framework for modularity on the web is probably one of the most important single factors that may change the way the construction of digital learning environments are viewed and constructed. In the first attempt made in the VWE project, there was no other way than making the environment proprietary. Even though it was possible to address the problem of information silos (in part) and to work with standards to a certain level, there was still not possible to build a digital learning environment that was completely open and based on standards. In the second attempt, in the WiLearn project, it was possible to work with the common concept of widgets-based mashups and a much more extensive use of standards both general technology standards, standards connected to the widget and mashup field and learning technology standards. Still there was a need to develop some quite extensive parts that could not be solely based on standards and common practise since the widget and mashup frameworks were not developed enough. From that perspective the situation in the WiLearn was a lot better than in the VWE project, but still far from ideal and there was still a need for quite a lot of tweaking and solutions developed by the project team.

From the experiences of the WiLearn project it can be concluded that the concept of widgets and mashups has matured and so has the related standards. The development of frameworks, such as the Wookie and Shindig frameworks have come far and are now fully usable and almost production stable. Even though the Apache Rave platform is still in early development it is reasonably stable. Much due to the fact that both Apache Wookie and Shindig that are used as the basis for Apache Rave are stable products.

From a technology point of view, the study worked well and there were only a few technical problems and some of them could be countered for in the field study as it was carried out in a controlled environment with members of the project team helping out. The problems that arose where mainly due to the prototype nature of the WiLearn platform, with limited and/or missing functionality or related to widgets developed for different versions of the standards. Some widgets found in the external widget stores were simply not able to run on the WiLearn platform. This was an annoying problem, but far from being a showstopper. Another limitation, related to the versioning problem, was the limited support for inter-widget communication. Inter-widget communication is essential for creating the same level of conceived integration as in a traditional LMS-system and for implementing advanced and rich functionality. In a mashup everything is loosely coupled and different widgets are normally completely self-standing and self-contained and not “aware” of the context in which they are used.

This makes it harder to maintain the feeling of a well-integrated learning environment. Still, the tight integration of LMS-functionality is often seen as a core problem of the LMS and is also what much of the criticism is targeting. With well-developed mechanisms for inter-widget communication integration can be managed by loose coupling instead, favouring adaptability and openness. There are strategies to deal with this problem and as the widget standards evolve, together with the implementation of html-feature, this problem is slowly being dealt with, se e.g. (Isaksson & Palmer, 2010) and (Sire, Paquier, Vagner, & Bogaerts, 2009) for an in-depth discussion on the inter-widget communication issue.

The WiLearn study, together with the experiences from previous studies, show that it is quite possible to implement all of the core functionality in a modular fashion using widgets and mashup technology. This basically means that mashup environments could potentially replace traditional LMSes. There are no restrictions regarding the type of digital learning environment that can be composed using a mashups. It is possible to create LMS-like learning environments as well as personal learning environments of combinations of the two. Everything depends on the widgets that are used in the mashup and the structure that is implemented. Hence, with the right set of widgets, a complete LMS could be built using as a mashup, even though an LMS may not be what is wanted or needed. The technical development is fast and by choosing modularity and standards in favour of monoliths, it is likely to be much more feasible to keep up with new and emerging technology. In fact, even though educational institutions in general are not among the early adopters of technology, many teachers are and everyone gains from an infrastructure for learning that is designed for change and that can meet with early adopters.

The study also confirms that modular learning environments are better adapted to suit different learning theories and pedagogical approaches as well as to different and changing pedagogical scenarios. Such features are beneficial and even essential in many learning scenarios, especially when working with pedagogical methods and approaches like Problem Based Learning (PLE) or case methodology, where it is hard (if not impossible) to foresee the learning path from start to finish beforehand – and thereby also to foresee the needs of the learning environment. Adaptability is also an essential property as it allows for the learning environment to be distributed (service-wise) over the Internet. At the same time it makes it possible to personalize and adapt the learning environment at the service level for group preferences as well as for personal preferences. The correlation between modular environments and adaptability and responsiveness, as well as technologies ability to match different learning theories, were discussed in detail in (Paulsson, 2008).

Several pedagogues also asked for ways to change the overall structure and look and feel of the WiLearn environment. This is an area that needs to be addressed in future versions.

Many pedagogues feel (and in fact are) restricted by the LMS and they got creative when presented to the mashup environment. Interesting, as well as a bit surprising, was that when pedagogues were given the freedom to influence how the learning environment was designed and what functions that should be available there was still a focus on the administration of learning and education. The difference was that the focus was on the small administrative tasks that is a part of the daily classroom activities and needed for communication with pupils and parent. This kind of support is often quite simple and surprisingly enough often missing in LMSes. Another, but related issue was the ability to create a personal "toolbox" with personal tools needed to facilitate teaching and all that is related. A common requirement was widgets that integrate different cloud services, such as Google Apps, DropBox, Twitter and likewise. There are widgets available that can do this, but the quality is varying and most of them are rather unsophisticated. The quality issue created a problem in general when searching widgets outside the WiLearn widget store. There is a huge amount of widgets available in different widget stores and many of them are of poor quality and they are adapted different versions of the standards, which cause some widgets not to function properly in the WiLearn environment. There is no good way of searching only for widgets intended for learning and education.

Another type of widgets that were asked for by the pedagogues participating in the study was support for specific pedagogical tasks and needs, such as widgets that could replace software for learners with special needs or widgets performing isolated tasks – i.e. replacing pedagogical desktop software. This is interesting and such widgets would in many cases be reasonably straightforward to develop. Unfortunately there are only a few widgets available of this kind and they are often dependent of language and cultural factors. In general, the supply of widgets that are especially developed for learning and education is small and of varying quality. Altogether, this contributes to an interesting business case where market competition is opened up for smaller actors to compete with LMS vendors by providing small and specialized components acting as building blocks in a mashup learning environments. In fact, it makes the learning environment independent of a specific LMS vendor to implement certain functionality. It has proven to be quite straightforward to develop simple widgets that can act as clients to different legacy (as well as to other) systems. The relation of system legacy and WiMupple is a concern that were raised a couple of times during the study and the idea of widgets acting as clients proved to be working, at least for simple cases.

In all, the study presented in this paper has contributed to the research field by providing a better understanding for how modularity and widget-based mashups influence the way that pedagogues approach the use of ICT and digital learning environments.

5 Future research

There are some potentially intriguing developments around the corner that are likely to benefit the development of mashup environments. On the one hand there is the general development, such as the gradual evolvement of html5 and standards for widgets and mashups. On the other hand there are developments within the field of learning technology standards, which at least on paper, look very promising from a modularity point of view. Among the most interesting efforts are the new specifications from IMS and especially the content standard IMS Common Cartridge (IMS CC) (2008) and the IMS Learning Tools Interoperability (IMS LTI) (2010) specifications. We are currently in the process of examining whether IMS CC can be used as a packaging format for WiLearn and furthermore, if IMS LTI can be used as a standard for widget communication and interactions related to pedagogical activities within a widget-based learning environment – as a complement to general standards. Severance et al. describes some experiments in (Severance et al., 2010) where IMS LTI was tested in a mashup environment and the results are promising and could be taken even further in the context of the WiLearn project.

The intention is to develop the WiLearn environment into a “production stable” framework for composing mashed-up digital learning environments that can be implemented and tested in a real-life educational scenario as a substitute to the LMS and to be able to study the use and effects in a real context over a longer period of time.

Another direction, that has already started, is the integration of the *Spider* and the WiLearn environment. The Spider is a national search service for digital learning resources that connects a number of repositories, using either metadata harvesting or federated search. This makes it possible to search for learning resources from several sources from a single point (Paulsson, 2009). The idea is to use the Spider as a widget store as well and by that making it possible to search for widgets and learning content in the same context.

References

Atwell, G. (2007). Personal Learning Environments - the future of eLearning? Retrieved 6 June, 2009 from www.elearningeuropa.info/files/media/media11561.pdf

Brereton, P., & Budgen, D. (2000). Component-based systems: a classification of issues. *Computer*, *33*(11). doi:10.1109/2.881695

Casquero, O., Portillo, J., Ovelar, R., Romo, J., & Benito, M. (2008). *iGoogle and gadgets as a platform for integrating institutional and external services*. Proceedings from Mash-Up Personal Learning Environments. Proc. of 1st Workshop MUPPLE.

Isaksson, E., & Palmer, M. (2010). *Usability and Inter-widget Communication in PLE*. Proceedings from Mashup Personal Learning Environments 2010, MUPPLE'10, Barcelona, Spain.

Erl, T. (2007). *SOA Principles of Service Design* (1 ed.). Boston, MA: Prentice Hall.

Fraternali, P., Rossi, G., & Sánchez-Figueroa, F. (2010). Rich Internet Applications. *Internet Computing*, *14*(3), 9-12. doi:10.1109/MIC.2010.76

Godwin-Jones, R. (2009). Emerging technologies personal learning environments. *Language, Learning & Technology*, *3*(9).

Gonzalez, M. A. C., Penalvo, F. J. G., Guerrero, M. J. C., & Forment, M. A. (2009). *Adapting LMS Architecture to the SOA: An Architectural Approach*. Proceedings from Fourth International Conference on Internet and Web Applications and Services. ICIW '09.

Häsel, M. (2011). Opensocial: an enabler for social applications on the web. *Communications of the ACM*, *54*(1), 139–144. Retrieved from http://doi.acm.org/10.1145/1866739.1866765

Hoyer, V., & Fischer, M. (2008). *Market Overview of Enterprise Mashup Tools*
Service-Oriented Computing ‚Äì ICSOC 2008. Proceedings from Lecture Notes in Computer Science.

IMS Common Cartridge Specification. Retrieved January 07, 2009 from http://www.imsglobal.org/cc/index.html

IMS GLC Learning Tools Interoperability Basic LTI Implementation Guide. *Version 1.0 Final Specification*, 25.

Jones, D. (2008). *PLES: FRAMING ONE FUTURE FOR LIFELONG LEARNING, E-LEARNING AND UNIVERSITIES*. Proceedings from Lifelong Learning Conference 2008, Yeppoon, Qld, AU.

Mitchell-Wong, J., Kowalczyk, R., Roshelova, A., Joy, B., & Tsai, H. (2007). *OpenSocial: From Social Networks to Social Ecosystem*. Proceedings from Digital EcoSystems and Technologies Conference, 2007. DEST '07.

Mulligan, C. E. A. (2009). Open API standardization for the NGN platform. *Communications Magazine, IEEE*, *47*(5), 108-113. doi:10.1109/MCOM.2009.4939285

Sultan, N. (2010). Cloud computing for education: A new dawn? *International Journal of Information Management*, *30*(2), 109-116. Retrieved from http://www.sciencedirect.com/science/article/pii/S0268401209001170

Nilsson, M. (2001). The Semantic Web: How RDF will change learning technology standards.

Ogrinz, M. (2009). *Mashup Patterns Designs and Examples for the Modern Enterprise.* Addison Wesley.

Palmér, M., Sire, S., Bogdanov, E., Gillet, D., & Wild, F. (2009). *Mapping Web Personal Learning Environments*. Proceedings from Mash-Up Personal Learning Environments (MUPPLE'09).

Paulsson, F. (2006). A Service Oriented Architecture-framework for modularized Virtual Learning Environments. In A. Mendes-Vilas, A. Solano Martin, J. Mesa Gonzáles, & J. A. Mesa Gonzáles (Eds.), *Current Developments in Technology-Assisted Education* (1 ed., p. 21 62). FORMATEX.

Paulsson, F. (2012). *Mashing up the learning environment - Evaluating a widget-based approach to Personal Learning Environments*. Proceedings from The Seventh International Conference on Internet and Web Applications and Services (ICIW 2012), Stuttgart.

Paulsson, F. (2008). *Modularization of the Learning Architecture: Supporting Learning Theories by Learning Technologies.* Stockholm: Royal Institute for Technology (KTH). Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-4712

Paulsson, F. (2009, May 24-28). *Connecting learning object repositories - strategies, technologies and issues*. Proceedings from The Fourth International Conference on Internet and Web Applications and Services, Venice, Italy.

Paulsson, F., & Berglund, M. (2006). *A Service Oriented Architecture-framework for modularized Virtual Learning Environments*. Proceedings from m-ICTE2006, Seville.

Paulsson, F., & Berglund, M. (2008). Suggesting a SOA-framework for modular virtual learning environments: comparing two implementation approaches. *International Journal of Web-Based Learning and Teaching Technologies*, *3*(1), 43-57.

Paulsson, F., & Berglund, M. (2003). SOA-frameworks for modular virtual learning environments: comparing implementation approaches. In J. Mosemann (Ed.), *Novel Approaches in Web-Based Learning Technologies: Tools for Modern Teaching.* Hershey, PA: IGI Global.

Paulsson, F., & Naeve, A. (2006a, Cunningham, Paul). *Establishing technical quality criteria for Learning Objects*. Proceedings from in the proceedings of eChallenges 2006, Barcelona, Spain.

Paulsson, F., & Naeve, A. (2006b). Virtual Workspace Environment (VWE): A Taxonomy and Service Oriented Architecture Framework for Modularized Virtual Learning Environments - Applying the Learning Object Concept to the VLE. *International Journal on E-Learning*, *5*(1), 45-57.

Preciado, J., Comai, S., & Sánchez-Figueroa, C. (2005). *Necessity of methodologies to model Rich Internet Applications*. Proceedings from the 2005 Seventh IEEE International Symposium on Web Site Evolution (WSE’05).

Salmenjoki, K., Paulsson, F., & Vainio, A. (2007). *Social Learning - How Web is Changing the Way We Process Information and Knowledge*. Proceedings from 5th International Conference on Emerging e-learning Technologies and Applications (ICETA), Stará Lesná, Slovak Republic.

Wilson, S., Daniel, F., Jugel, U., & Soi, S. (2012). Orchestrated User Interface Mashups Using W3C Widgets. In A. Harth & N. Koch (Eds.), *Current Trends in Web Engineering* (pp. 49-61). Springer Berlin Heidelberg.

Severance, C., Hanss, T., & Hardin, J. (2010). IMS Learning Tools Interoperability: Enabling a Mash-up Approach to Teaching and Learning Tools. *Techn., Inst., Cognition and Learning*, *7*, 245-262.

Sire, S., Paquier, M., Vagner, A., & Bogaerts, J. (2009). *A messaging API for inter-widgets communication*. Proceedings from the 18th international conference on World wide web, Madrid, Spain.

Anderson, T., & Shattuck, J. (2012). Design-Based Research : A Decade of Progress in Education Research? 41: 16-25,http://edr.sagepub.com/content/41/1/16.full. *Educational Researcher*, *41*, 16-25.

Varlamis, I., & Apostolakis, I. (2006). The present and future of standards for e-learning technologies. *Interdisciplinary Journal of Knowledge and Learning Objects*, *2*, 59–76.

Verpoorten, D., Westera, W., & Specht, M. (2011). Infusing reflective practice in eLearning courses – can widgets help? *International Journal of Technology Enhanced Learning*, *3*(1), 93-109. Retrieved from http://dx.doi.org/10.1504/IJTEL.2011.039066

Wang, F., & Hannafin, M. (2005). Design-based research and technology-enhanced learning environments. *Educational Technology Research and Development*, *53*(4), 5-23. Retrieved from http://dx.doi.org/10.1007/BF02504682

Wheeler, S. (2009). Learning Space Mashups: Combining Web 2.0 Tools to Create Collaborative and Reflective Learning Spaces. *Future Internet*, *1*(1), 3–13. doi:10.3390/fi1010003

Wilson, S., Blinco, K., & Rehak, D. (2004). *An e-Learning Framework A Summary*. Proceedings from Alt-I-lab 2004.

Wilson, S., Liber, O., Johnson, M., Beauvoir, P., Sharples, P., & Milligan, C. (2006). *Personal Learning Environments: Challenging the dominant design of educational systems*. Proceedings from First European Conference on Technology Enhanced Learning (ECTEL), Crete, Greece.

Wilson, S., Sharples, P., & Griffiths, D. (2008). *Distributing education services to personal and institutional systems using Widgets*. Proceedings from Mashup Personal Learning Environments (MUPPLE08), Maastricht.

Wong, J., & Hong, J. (2008). *What do we" mashup" when we make mashups?* Proceedings from the 4th international workshop on End-user software engineering.

(2011). Yahoo! Query Language. Retrieved from http://developer.yahoo.com/yql/

Yu, J., Benatallah, B., Casati, F., & Daniel, F. (2008). Understanding Mashup Development. *IEEE Internet Computing*, *12*(5), 44-52. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=eoah&AN=17067655&site=ehost-live&scope=site