

Article

Identifying Middlewares for Mashup Personal Learning Environments

Jinan Fiaidhi*, Sabah Mohammed, Lyle Chamarette and David Thomas

Department of Computer Science, Lakehead University, Thunder Bay, Ontario P7B 5E1, Canada; E-mails: mohammed@lakeheadu.ca (S.M.); lfchamar@lakeheadu.ca (L.C.); dathomas2@lakeheadu.ca (D.T.)

* Author to whom correspondence should be addressed; E-mail: jfiaidhi@lakeheadu.ca; Tel.: +1-807-343-8224; Fax: +1-807-766-7243

Received: 29 July 2009; in revised form: 31 July 2009 / Accepted: 3 August 2009 /

Published: 5 August 2009

Abstract: The common understanding of e-learning has shifted over the last decade from the traditional learning objects portals to learning paradigms that enforces constructivism, discovery learning and social collaboration. Such type of learning takes place outside the formal academic settings (e.g., seminars or lectures) where a learning environment is created by using some kind of web application mashup tools. The use of these mashup tools moves the learning environment further away from being a monolithic platform towards providing an open set of learning tools, an unrestricted number of actors, and an open corpus of artifacts, either pre-existing or created by the learning process – freely combinable and utilizable by learners within their learning activities. However, collaboration, mashup and contextualization can only be supported through services, which can be created and modified dynamically based on middlewares to suit the current needs and situations of learners. This article identifies middlewares suitable for creating effective personal learning environment based on Web 2.0 mashup tools. This article also proposed a general framework for constructing such personal learning environments based on Ambient Learning realized by learning agents and the use of Enterprise Mashup servers.

Keywords: enterprise mashup; personal learning environment; Web 2.0; ambient learning; learning agents

1. Introduction: Understanding the Nature of Mashup Services

The emerging paradigm of Web 2.0 is transforming traditional Internet from a mass media to a social media mode. The fundamental communication premise in social media mode is one-to-one personal communication. This type of new communication style revolutionized learning and identified new breed of learners. According to Hart [1], the new type learners are identified through the following features:

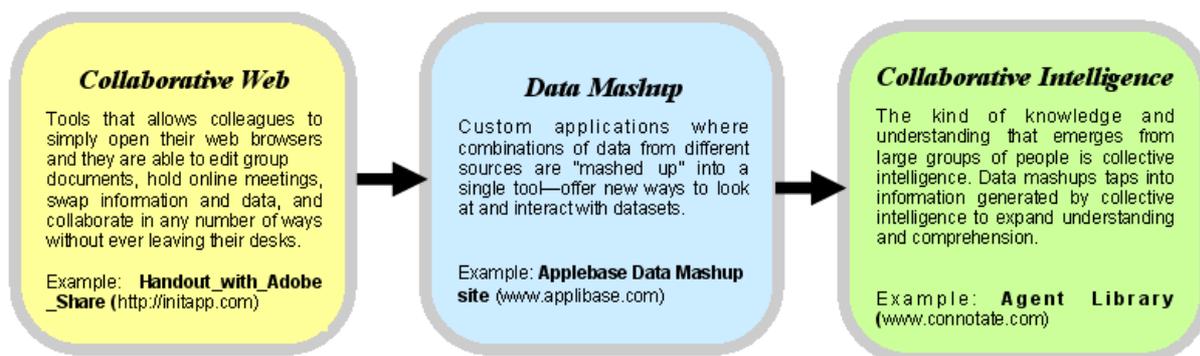
- Learners who prefer hyperlinked information that comes from many sources.
- Learners with multiple skilled and interacting with others.
- They are highly visual learners, preferring to process pictures, sounds, and video rather than text.
- They are experiential learners who learn by discovery rather than being “told.”
- They are very social, and love to share with others. They enjoy working in teams.
- They are happy to take on different roles in their learning (e.g. student, facilitator).
- They prefer to learn “just in time.”
- They need immediate feedback, responsiveness and ideas from others
- They are very independent learners;
- They prefer to construct their own learning by assembling and mashing information.

However, “placing such learners and relationships at the center of informational space will have a profound influence at all levels of academia. It will change the way we relate to knowledge and information; the way we do research and evaluate credibility; the way educators and students interact with each other; and the way students learn to be professionals in their chosen disciplines” [2]. In Web 2.0 paradigm, applications are delivered through Web browsers and the applications are responsive to users through rich application interfaces, often including pre-built application components or service widgets. Mashups are the essence of such applications. A mashup is a website, a service or an application that combines content from more than one source into an integrated product. Today mashups are very common as the new authoring tools and middlewares are being developed that enables building new products without much programming. As tools and middlewares like these become more robust, we will see increasing use of mashups in teaching and learning. There are many mash-up applications, such as map mash-up (Yahoo! Pipes (<http://pipes.yahoo.com/pipes/>), Google Earth (<http://earth.google.com/>)), Video Mash-Up (www.wikieducator.org) and News Mash-Up (www.feedchronicle.com). However, how to mash up information effectively is a challenging issue. For this reason, the 2008 Horizon Report [2] classifies mashups into three categories with increasing level of semantics awareness. Figure 1 illustrates these categories.

Regardless of the mashup category is, it generally boils down to the following architectural issues:

- Accessing the enterprise services and data repositories,
- Using Web-based mashups (e.g. using Software as a Service (SaaS)),
- Using User-friendly assembly models,
- Using mashup management and maintenance module,
- Using modules for enterprise mashup security.

Figure 1. Increasing Levels of Mashup Semantic Awareness.



These issues simply mean that any mashup service must be based on some sort of content aggregation technologies. The traditional content aggregation technology was until recently based on *Portals* and *Portlets*. Portals are designed as an extension to traditional dynamic Web applications, in which the process of converting data content into marked-up Web pages is split into two phases - generation of markup "fragments" and aggregation of the fragments into pages. Each markup fragment is generated by a "portlet", and the portal combines them into a single Web page. Portlets may be hosted locally on the portal server or remotely on another server. However, the portal technology is about server-side, presentation-tier aggregation and it cannot be used easily to drive more robust forms of application integration. Mashup services are more loosely defined where content aggregation can take place either on the server or on the client side. The base standards for the mashup content aggregation are the XML interchanged as REST (REpresentational State Transfer) or Web Services. Lightweight protocols for content aggregation such as RSS (Really Simple Syndication) and Atom are commonly used for the purpose of facilitating the aggregation of contents. The major difference between portals and mashups is in the way content or services composition is administered. Portals achieve composition through the use of application server as a mediator between a client browser and services. Mashups, however, perform composition directly from the end user's browser. Although the client-side architecture helps mitigate a performance problem by avoiding the use of the application server, it can introduce its own problems in the design, use and maintenance of mashup applications where often such application users find participating in mash-up process as time-consuming or even a trouble. To address this problem, mashup middlewares provide better support for the users to mash-up or enjoy the result of mash-up. Certainly mashup middlewares have been used for promoting Personalized Learning Environments (PLEs) [3], where it helps the learners in utilizing open and heterogeneous set of learning tools to connect with each other into a learning network and to collaborate with other actors on shared artifacts along different learning activities [4]. The pedagogical motivations behind using mashups within PLEs are obvious as learners have different learning styles, background knowledge, interests and alternative evolution of their learning over time, which makes ineffective offering the same packaged course to each learner. Learning should be a user-centred process focused on learning activities adapted to learner's needs and preferences [5]. Experiments on using mashups for the MUPPLE PLE [6] in writing a collaborative article, for example, showed quite promising success in creating a comprehensive learning environment. There reasons for such success compared to the traditional instructional techniques can be summarized as follows:

- There is no ‘perfect’ instructional designer: an environment can only be planned for the average learner, not the individual. Learners need to actively adapt their learning environment to their needs so that they can construct the competences necessary for successful learning. And facilitators can coach them on this way. Learners proactively have to take account of their learning environment [6].
- Learners can develop the judgments and skills or literacy necessary for using new technologies in a rapidly changing society [7]. A response to pedagogic approaches which require that learner’s e-learning systems need to be under the control of the learners themselves [8].
- Mashups enables web services to be combined much more tightly so that data from one is used to access information in another. This will enable learners to create larger learning services and applications [9].

PLEs are not an entity, structural object or software program in the sense of a learning management system. Essentially, they are a collection of tools, brought together under the conceptual notion of openness, interoperability, and learner control. As such, they are comprised of two elements - the tools and the conceptual notions that drive how and why we select individual parts. The tools are mainly Web 2.0 tools supporting collaborative and social networking services. The conceptual notions are what users try to achieve with offered services. Hence, PLEs are a concept-entity and they are very personalized and individual. Compared with the traditional Virtual Learning Environments (VLEs) where most of the educational institutions use it for their formal learning, PLEs provide the new level of learning that is required by the new type of learners. Although there are many models for implementing PLEs where mashups can be one of its supporting services (e.g. Jafari [10], PLEF [11], PLEX [12], ELGG (<http://elgg.org/>), Webtop [13]), such models require a middleware to enable collaboration and mashup services. In the next section, the article shed some light on the notion of mashup middlewares.

2. Mashup Middlewares

A mashup middleware can have a *primitive* or *composite* architectural structure. The most primitive mashup middleware contains only a feed reader. For example, Google Reader (www.google.com/reader) is a primitive mashup middleware that can follow the feeds from other web sites and inform the user for any news transmitted in the feed (i.e. latest information added on some web site via RSS or Atom). One can insert feeds in Google Reader and later read the news from there without the need to go to the specific web site. Google Reader is very useful middleware because it saves mashup clients lot of time and delivers information which the client has interested in. Even if in case that the provided information is just too much, one can use tags or folders to organize such provided information as Google Reader enables the user to assign such tags to different feeds, so that the information can have some structure and is easily perceived. However, one can build on this primitive mashup middleware, for example by adding some sort of context and makes the feed reader aware of the user mashup needs. Many mashup middlewares just try to do that (e.g. the PLATE System [14]). A very simple example is provided by Mitko Denev [15] where he used Google Reader in conjunction with Netvibes aggregator (www.netvibes.com). In this direction, you need first to set up your start page in Netvibes and it creates tabs which correspond to different contexts such as work,

home, hobby and web applications. Then insert your Google Reader feeds at the appropriate tabs. In this way you solved both the problem associated with the linear presentation of information as well as the lack of context. Netvibes allows you to arrange the elements of the start page spatially. Reading the feeds in context helps you filter, assimilate and remember the useful items. However, not all the composite middlewares are as simple as Google Reader-Netvibes. Most commonly, composite middlewares represent an *Enterprise Mashup Infrastructure* with the following components:

- Services and information produced in a format that can be mashed, such as RSS/Atom, Web services, or REST
- Visualization components such as portlets, widgets, and gadgets
- A tool to compose the mashup and wire components together
- A robust infrastructure to manage all the change and new combinations

Examples of Enterprise Mashups include JackBe (www.jackbe.com), Mainsoft SharePoint Federator and IBM WebSphere Portal (www.mainsoft.com), Oracle Fusion Middleware (www.oracle.com/middleware), WSO2 Middleware (<http://wso2.com>), Kapow Mashup Server (www.kapowtech.com), SnapLogic (www.snaplogic.org) and Apache XAP (<http://incubator.apache.org/xap/>). Generally, there are two main components that can be identified at any Enterprise Mashup infrastructure: Mashup builder and Mashup Enabler. The mashup builder produces the user interface of the resulting mashup through connecting several required widgets to create a composite application. The mashup enabler accesses the mashed-up unstructured data and makes the internal and external resources available based on REST, RSS or Atom. Both components may be called an Enterprise Mashup Server. Figure 2 illustrates the way the Enterprise Mashup Server works as a middleware and Figure 3 illustrates the general structure of Enterprise Mashup Server.

The 2008 Horizon Report [2] calls the Enterprise Mashup as the Social Operating System as it represents the basic organization of the network around people, rather than around content. However, the architecture of such social operating system becomes more complex as the degree of semantic awareness increases as well as with the increasing freedom of social connectivity ((e.g. across enterprises as in the *Cloud paradigm* [16]).

Figure 2. Enterprise Mashup Server as a Middleware.

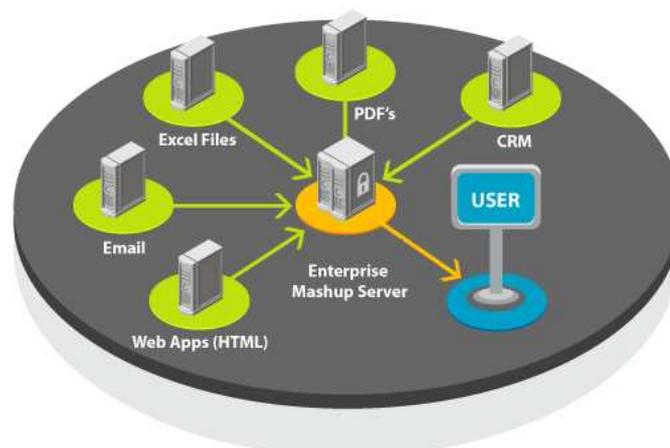
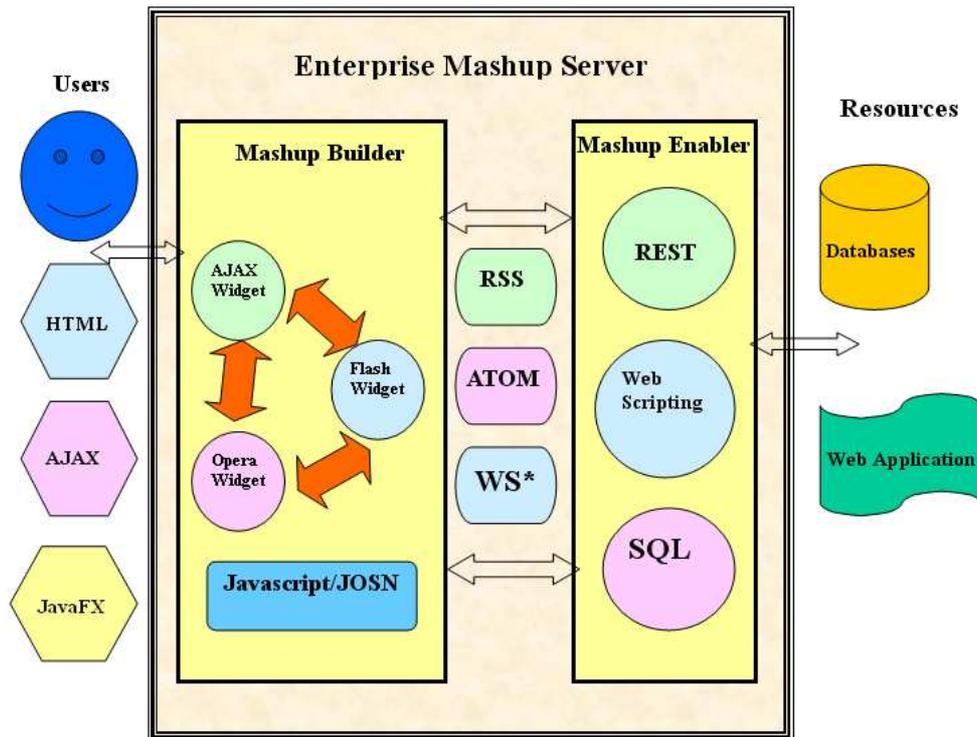


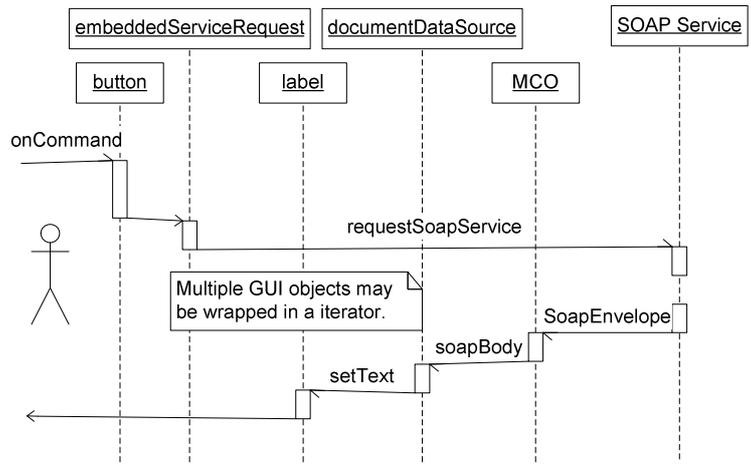
Figure 3. The Main Components of Enterprise Mashup Server.



2.1. Experimenting with the Apache XAP Enterprise Mashup Server

XAP (Extensible AJAX (Asynchronous JavaScript and XML) Platform) provides an XML-based framework for building, deploying and maintaining rich, interactive, Ajax-powered web applications. A basic principal of XAP is to leverage existing Ajax projects such as Dojo (i.e. providing primitives for supporting OpenAjax and its secure mashup services), as well as other community efforts such as Eclipse ATF (Ajax Toolkit Framework). Completely eXtensible, XAP can plug-in any Ajax toolkits or you can create your own components to extend the markup language. By using XAL, Nexaweb's declarative markup language, the amount of scripting required is dramatically reduced, solving key development and maintenance challenges associated with Ajax programming. XAP can also be considered as an Enterprise Mashup Server. XAP is a pure client-side runtime that is implemented using Ajax. It supports popular modern browsers and can work with any server side environments. XAP defines presentation and data binding in XML along with client side event handling using what is called MCOs (Managed Code Objects) code. An MCO code is used to reference functions in Javascript or JSP files when XAP inspect the programming code. Basically what MCOs tries to do is to connect to various servers via the SOAP protocol. Figure 4 illustrates how the user communicates with the MCOs via the user application interface.

Figure 4. Communication through XAP Rich Client.



However, you do not need to understand all the programming details of writing XAP programs and communicating with MCOs if you use the Nexaweb Studio (www.nexaweb.com), which provides seamless XAP project creation as well as full control of the code and functionality. Figure 5 illustrates the Nexaweb Studio where it has been used to create an XAP application (called Get-Weather) to check the weather conditions of a given city. When the “Get Weather” button is clicked the XAP application builds a SOAP message based on the given city name provided in the text box and send it to the city MCO. When SOAP response is received the display is updated using the XAP OpenAJAX where only the weather information area is refreshed (see Figure 6 (a)). However, we can add more to this textual result by mashing-up the map of the city on an area next to the weather textual results. For this purpose, we can press the “Generate Map Link” button and similarly by providing the latitude and longitude of the city, through SOAP and the relevant MCO code, XAP builds a link with that information (see Figure 6 (b)) and accordingly a window is opened to show the city map through Google Maps.

Figure 5. Using the Nexaweb Studio to Design a Weather Mashup Application.

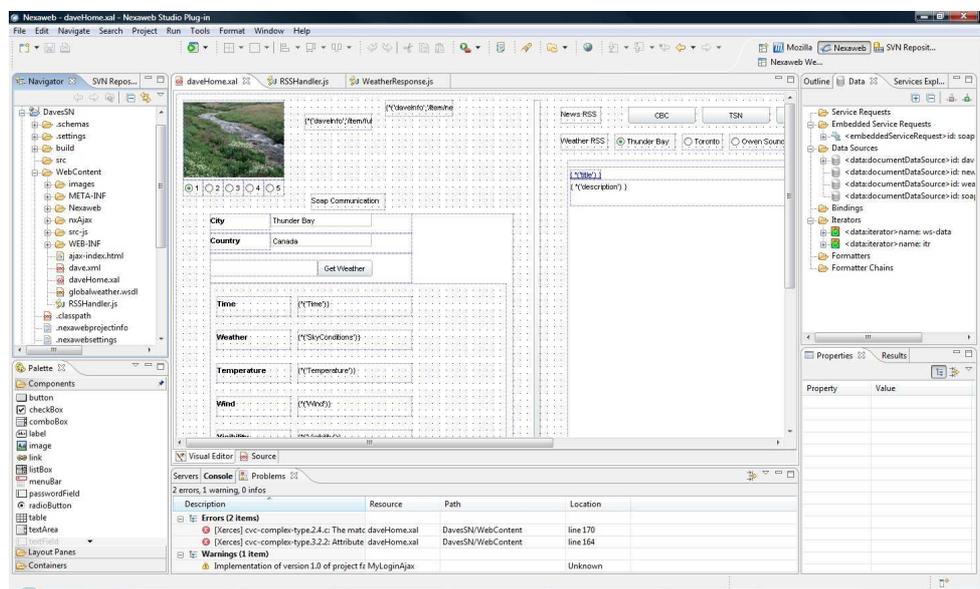
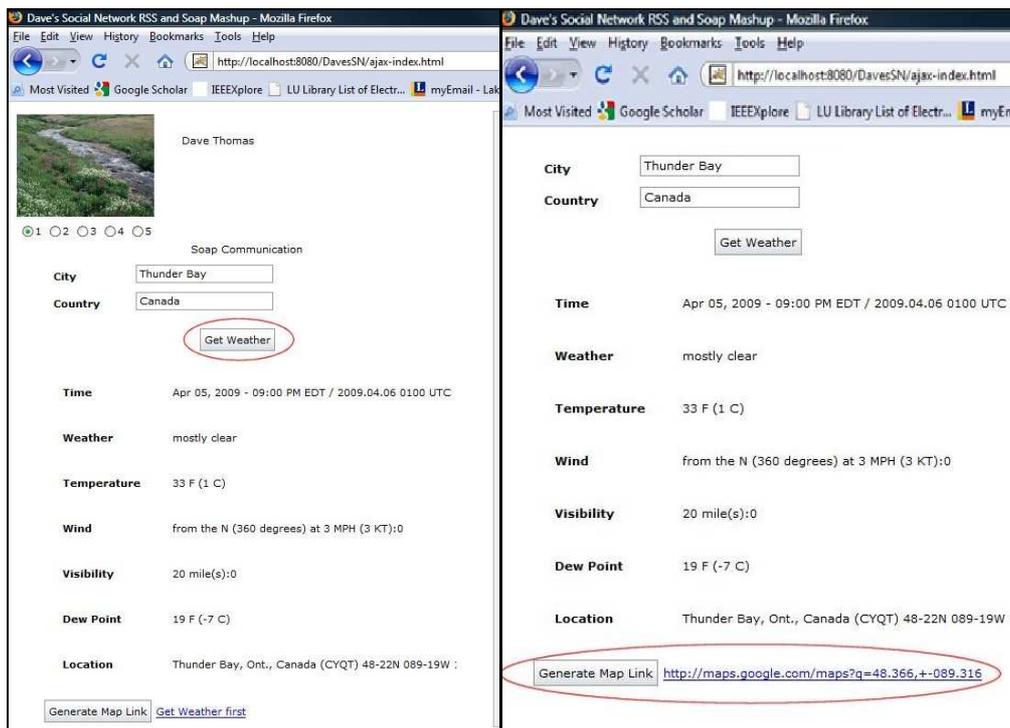


Figure 6. Mashing up the Weather Information then the City Map.



(a) After Pressing Get Weather

(b) After Pressing Generate Map Link

However, you can follow two easy tutorials from Nexaweb which can enable you to build your similar mashup applications:

Nexaweb Tutorial 1: <http://www.nexaweb.com/home/us.dev/index.html@cid=1996.html>,

Nexaweb Tutorial 2: <http://www.nexaweb.com/home/us.dev/index.html@cid=1989.htm>.

3. Towards Using Mashups for Personal Learning Environments

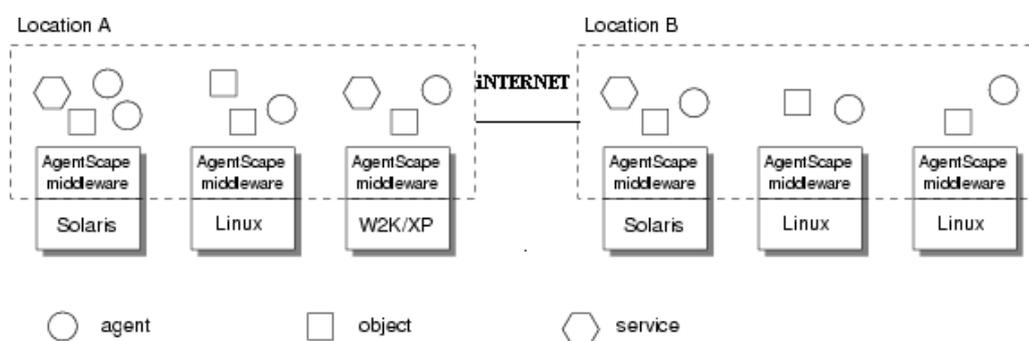
Learning plays an ever increasing role in our lives with the growing need for knowledge acquisition and the faster information overload cycles. For centuries, we have considered it appropriate to decouple learning from daily life and to emphasize it in specialized educational institutions. Increasingly, we find it important to re-connect learning with daily life: learning happens at freely chosen times and locations, in the work-life and daily-life processes which stimulate us to learn: learning becomes ambient. Ambient Learning and Ambient Intelligence are promising concepts for new methods of learning and in particular for adapted, comprehensive and personalized learning environments [17]. These concepts help learners and institutions to keep up with the rapidly changing knowledge-based economy. Ambient learning is designed to facilitate access to high quality e-learning material tailored to the needs of an individual learner. These needs are determined by the time, place, pace and context that best suits the individual learner. Ambient learning through the provision of content integration and composition allows access to, new e-learning material as well as existing catalogues/e-learning resources [18]. Although ambient learning is based on Ambient Intelligence Technologies (AmI) [19] its use is not only limited to rooms and buildings. Ambient learning is taking new dimensions as it can be realized by technologies that combine both Ambient Intelligence and collaborative learning through

Web 2.0 technologies. The basic advantage of using AmI in any application is to enhance interactions between objects. However, the social focus/perspective of the AmI research is largely neglected. Instead of enhancing interactions with technological objects, there is a need for possible AmI applications that can enhance interactions with other people [20]. Enhancing the interactions among people contributes to what is currently termed as the Social Ambient Intelligence or Ambient Awareness [21]. Actually, the notion of ambient awareness has its roots in ideas of constructivism and discovery-learning [22]. Constructivism and Discovery learning is an inquiry-based, constructivist learning theory that takes place in problem solving situations where the learner draws on his or her own past experience and existing knowledge or relationships to discover facts and relationships and new truths to be learned. Although constructivism is not a particular pedagogy, it can be used to promote active learning if it can be associated with some sort of intelligent discovery like AmI or ambient awareness. However, central to the ambient awareness research is the use of Web 2.0 techniques. In fact, the Web 2.0 approach has revolutionized the way we use the web and certainly, it can have major positive impact on the AmI research. On one hand, Web 2.0 enables the active participation of users with new contents such as wiki pages, blogs or online multimedia. On the other hand, Web 2.0 transforms the Web into an application-enabling platform. *Enterprise Mashups*, one of the hottest Web 2.0 technologies today, could affect your ambient learning in a very positive way. Before the enterprise mashup, the user had to sign in to several applications and go to different Web sites to manually collect the information and then try to make sense of it. The enterprise mashup web application overcomes this hassle in an elegant way and allows the user to harness more of the collective intelligence in the enterprise to make better decisions. Mashups have the opportunity to increase the strategic value of learning—by delivering enriched information to users—and reduce time cycles spent on custom development. However, mashups have been around for years and the concept of the end-user being able to easily ‘drag-drop’ and put together a hacked up application within minutes has not become a reality- and this is what is causing a major obstacle. For mashups to really take off, we need to be able to capture the context of information. Information becomes relevant and more useful when it is placed in the right context. If a Mashup can leverage of some form of social context, it would then be able to provide the relevant information to the user. In this direction, social mashups are a new trend that takes the traditional mashup one-step further. Ultimately, in an enterprise, social interaction is a key part of how information is tied together and increasingly more relevant to how individuals want to visualize information. Hence, linking people, processes and information through mashing can create a real social enterprise mashup. On one hand, mashups by their very definition, involve a man-in-the-middle and rely on RESTful communication protocols. On the other hand, ambient learning aims at seamless delivery of ubiquitous services, continuous communications and intelligent user interfaces and context-awareness. In this sense, ambient learning systems need to provide autonomy, distribution, adaptation, pro-activeness and responsiveness as the key characteristics, which are similar to the characteristics learning agents [23]. Learning Agents are computer programs capable of flexible autonomous actions in a dynamic environment and are apparently a suitable choice for implementing ambient intelligence systems [24]. In fact, learning agents inside the Enterprise provide an ecosystem for creating and sharing learning knowledge [25]. Based on learning agent’s solutions, learners are empowered with personalized software assistants or learning agents to uncover high-value data, resulting in cost

reduction and higher productivity. Using Learning Agents "trained" to do anything a human can to monitor, harvest, extract, process, deliver and integrate dynamic content from the internet, intranets, extranets and Enterprise applications – Learners can access the data that are normally inaccessible. Not only can information be accessed, it can be shared and mashups can be created and made available to key learners. Empowering learners in this way can produce enormous educational returns.

To create an ecosystem of learning agents, one only needs to find the suitable toolkit that can create such collaborative learning environment. There are many multi-agents toolkits available these days, e.g. JADE (<http://jade.tilab.com/>), but most of them do not support interoperability between agents working at different locations across the web. This criteria is very important when such agents need to perform mashing-up services across multiple Web locations based on tools like the XAP. For this purpose, we found the AgentScope (<http://www.iids.org/research/aos>) is the suitable toolkit for creating interoperable learning agents capable of performing mashup services. Figure 7 illustrates an abstract view on the interoperability principle build through AgentScope.

Figure 7. AgentScope Interoperability Principle.



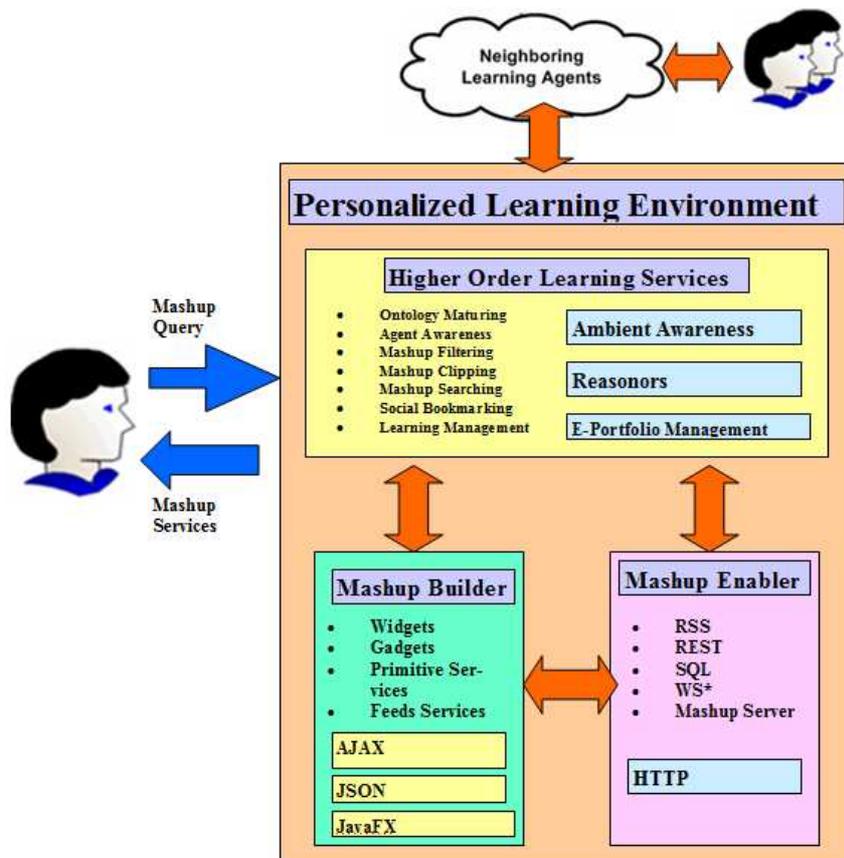
Basically, the AgentScope is a middleware that provides a kernel and a number of system services (e.g. agent servers, location managers, and host managers). The agent server gives an agent access to the AgentScope middleware layer. A location manager is the coordinating entity in an AgentScope location (thus managing one or more hosts in one location). Agent creation, migration, and all kinds of policy related issues that are relevant in the context of a location, is managed or coordinated by the location manager. The host manager manages and coordinates activities on a host at an AgentScope location. The host manager acts as the local representative of the location manager, but is also responsible for local (at the host) resource access and management. At each location there can be multiple agents, services, and hosts. Each location runs a number of services; there is one Location Manager (LM) per location and it is responsible for keeping track of the hosts within that location. For each host in a location there will be a corresponding Host Managers (HM) which is responsible for managing the resources on that host. For each supported type of host there is an Agent Server (AS) that is responsible for running the actual agents. A Lookup Service maps names to their addresses. And lastly the Directory Service is responsible for mapping services to names. In order to layout an AgentScope learning ecosystem, the locations need to be created first. Once a server has been selected as a location, the Location Manager must be setup. Next hosts need to be started accompanied by a Host Manager for each type of host. Once the Host Manager is running then agents can be installed on the location; AgentScope will automatically determine the correct Host Manager to register the agent

with. When all the necessary components are started and registered the agents will be able to communicate with each other. You do not need to know much details about the AgentScape middleware, one only need to experiment with the AgentScape tutorial demonstration (<http://www.iids.org/research/aos/publications.html>) and that will provide the basic knowledge for building your own learning ecosystem.

4. Conclusions

Although mashup holds great promises, it is still not clear when and how the existing systems will exploit this new learning model. This article advocates for the use of mashup middlewares to help in remixing, reusing, adapting and personalizing information where such issues play an important role in any learning systems by enabling learners to learn based on their needs. The basic building blocks of a mashup middleware are the mashup builder and mashup enabler. This article introduced an example on such middleware using the Apache XAP Enterprise Mashup Server. However, this article identified one more mashup middleware (AgentScape) which supports greater interoperability if mashing-up are required to be achieved across different paradigms and enterprises. Moreover, we can argue that one can add many other components to these two basic components of the mashup middleware to produce more sophisticated and intelligent personal learning environment. This article identified the importance of including AmI and ambient learning as one additional component to provide an adaptive and personalized learning. To draw an enhanced image of the mashup middleware framework involving this new component, we used figure 8 to help in illustrating our vision. The proposed framework utilizes learning agents where users can interact on a level that best suits their needs and capabilities, leaving tedious chores to the learning agents. Ambient learning or awareness enables the learning agents to learn about their user and adept the environment. Mashups on such environment increases the strategic value of learning—by delivering enriched information to users—and reduce time cycles spent on custom development. Actually the ambient awareness is part of larger higher-order learning services that include some basic mashup support services (e.g. filtering, searching, clipping and bookmarking) as well as some advanced mashup support services (e.g. learning management, mashup ontology maturing and e-portfolio management). Actually ambient awareness provides the reasoning and logic for enabling the advanced mashup support services. Suppose our goal is to develop a PLE system that can help users to identify mashups based on their associated tagging. This scenario requires that the PLE needs to develop certain ontology for mashups which may not exist at the initial learning process. For this purpose, the PLE system needs to have a component for ontology maturing to enable the proper identification of the required mashups. Such ontology maturing may pass into various awareness phases: starting with introducing simple mashup tags where each learner need to contributes to the collaborative development of mashup ontology [26]. Thus, each community member can contribute new ideas (tags) emerging from the usage to the development of ontology (phase I “Emergence of Ideas”). The community picks them up, consolidates, and refines them (phase II “Consolidation in Communities”) and formalizes with semantic relations towards a lightweight ontology (phase III “Formalization”) or even adds axioms for improving inference processes (phase IV “Axiomatization”). In this way the learners themselves can directly execute changes if needed.

Figure 8: A framework for Mashup Personalized Learning



The first two authors are currently engaged in developing a prototype for implementing the proposed enhanced mashup middleware. Some early references to the progress in implementing various aspects of the ambient awareness part of this middleware can be found in [27, 28].

Acknowledgements

The first author’s NSERC Discovery Grant supports this research.

References

1. Hart, J. Understanding Today’s Learner, Learning Solutions e-Magazine, September 22, 2008, <http://www.c4lpt.co.uk/articles/understandinglearners.pdf>.
2. The 2008 Horizon Report, The New Media Consortium and EDUCAUSE, ISBN 0-9765087-6-1, www.nmc.org/pdf/2008-Horizon-Report.pdf.
3. Lamb, B. The Urgency of Open Education, The MetaMedia Blog, June 2009, <http://metamedia.typepad.com/metamedia/2009/06/brian-lambs-the-urgency-of-open-education.html>.
4. Redecker, C. Review of Learning 2.0 Practices: Study on the Impact of Web 2.0 Innovations on Education and Training in Europe, EUR 23664 EN – 2009, <ftp://ftp.jrc.es/pub/EURdoc/JRC49108.pdf>.

5. Santos, O.C.; Boticario, J.G. Meaningful pedagogy via covering the entire life cycle of adaptive eLearning in terms of a pervasive use of educational standards: The aLFanet experience. In *ECTEL 2006, LNCS*; Tomadaki, E.; Scott, P., Eds; Springer: Heidelberg, 2006; Volume 4227, pp. 691-696.
6. Wild, F.; Mödritscher, F.; Sigurdarson, S. Designing for Change: Mash-Up Personal Learning Environments, *eLearning Papers Journal*, N° 9, July 2008, ISSN 1887-1542, <http://www.elearningeuropa.info/files/media/media15972.pdf>.
7. Attwell, G. Personal Learning Environments - the future of eLearning?, *eLearning Papers Journal*, Vol 2, N° 1, January 2007, ISSN 1887-1542, <http://www.elearningeuropa.info/files/media/media11561.pdf>.
8. Harmelen, V. M. Personal Learning Environments. Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies, ICALT 2006, Kerkrade, The Netherlands, July 2006; pp.815-816.
9. Lubensky, R. The present and future of Personal Learning Environments (PLE, Deliberations Blog, December 2006, <http://www.deliberations.com.au/2006/12/present-and-future-of-personal-learning.html>.
10. Jafari, A.; McGee, P.; Carmean, C. Managing Courses, Defining Learning: What Faculty, Students, and Administrators Want, *EDUCAUSE Review*, Volume 41, Number 4, July/August 2006, pp. 50-71.
11. Chatti, M. A. Personal Environments Loosely Joined, BlogSpot, January 2007, <http://mohamedaminechatti.blogspot.com/2007/01/personal-environments-loosely-joined.html>.
12. Wilson, S. PLEX, Experiences in building a composite application, www.cetis.ac.uk/members/ple/resources/edf.ppt.
13. Martin, M. My Personal Learning Environment, TypePad Blog, April 2007, http://michelemartin.typepad.com/thebambooprojectblog/2007/04/my_personal_lea.html.
14. Li, Y.; Fang, J.; Xiong, J. A Context-Aware Services Mash-Up System, Seventh International Conference on Grid and Cooperative Computing, 2008; pp.707-712.
15. Denev, M. Google Reader as Middleware, Trends in the Web Blog, May 4, 2008, <http://web-inside.blogspot.com/2008/05/google-reader-as-middleware.html>.
16. Siebeck, R.; Janner, T.; Schroth, C.; Hoyer, V.; Wörndl, W.; Urmetzer, F. Cloud-based Enterprise Mashup Integration Services for B2B Scenarios, In 2nd Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009) in conjunction with the 18th International World Wide Web Conference (WWW2009), Madrid, Spain, 2009.
17. Paraskakis, I. Ambient Learning: Rationale and its use in Supporting Blend Learning for Executive Education. Sixth International Conference on Advanced Learning Technologies, ICALT 2006, Kerkrade, The Netherlands, July 2006; pp. 144-146.
18. Pawlowski, J.M.; Bick, M.; Veith, P. Context metadata to adapt Ambient Learning Environments, 2nd IEEE International Interdisciplinary Conference on Portable Information Devices, Germany, August 2008, pp. 1-6.
19. Piquepaille, R. Capturing ambient intelligence, ZDNet Online Magazine, 15 January, 2008, <http://blogs.zdnet.com/emergingtech/?p=804>.

20. Cassens, J. Explanation Awareness and Ambient Intelligence as Social Technologies, Ph.D. Thesis, Norwegian University of Science and Technology, Faculty of Information Technology, Mathematics and Electrical Engineering, Department of Computer and Information Science, 2008.
21. Rizopoulos, C. An activity-based perspective of interacting with ambient intelligence systems, 3rd IET International Conference on Intelligent Environments (IE 07), Germany, September 2007; pp. 81-88
22. Bruner, J.S. *On knowing: Essays for the left hand*. Harvard University Press, Cambridge, USA, 1967.
23. Hagaras, H.; Callaghan, V.; Colley, M.; Clarke, G.; Pounds-Cornish, A.; Duman, H.. Creating an Ambient-Intelligence Environment Using Embedded Agents. *IEEE Intelligent Systems* **2004**, *19*, 12-20.
24. Corchado, J.; Tapia, D.; Bravo, J. 3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008, Advances in Intelligent and Soft Computing Series, Salamanca, Spain, October 2008; Springer-Verlag, 2009; Volume 51.
25. Vuori, E.K. Knowledge-intensive service organizations as agents in a business ecosystem. In Proceedings of ICSSSM '05, 2005 International Conference on Services Systems and Services Management, Chongqing, China, June 2005, Volume 2, pp. 908 – 912.
26. Braun, S.; Schmidt, A.; Walter, A.; Zacharias, V. The Ontology Maturing Approach for Collaborative and Work Integrated Ontology Development: Evaluation Results and Future Directions. Presented at ESOE2007, Busan, Korea, November 2007.
27. Mohammed, S.; Fiaidhi, J.; Mohammed, O. Sharing Biomedical Learning Knowledge for Social Ambient Intelligence. *Journal of Computers* **2009**, *4*, 905-912.
28. Mohammed, S.; Fiaidhi, J.; Mohammed, O. Developing an Ontology Extraction Agent for a Biomedical Learning Social Network, Presented at NASTEC 2008 International Conference, Montreal, Canada, August 2008.

© 2009 by the authors; licensee Molecular Diversity Preservation International, Basel, Switzerland. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).