Design and Implementation of a Cloud-based Personal Learning Environment

Jose Luis Roberto Asuncion, University of the Philippines - Diliman, Philippines
jose.asuncion@gmail.com
Michelle Lee Moscatel, University of the Philippines - Diliman, Philippines -
michelle.moscatel@gmail.com
Rommel P. Feria, University of the Philippines - Diliman, Philippines -
rpferia@dcs.upd.edu.ph

Abstract
While traditional approaches to learning caters to institutions, other models of learning are more suitable to a learner's needs in different situations. Rapid advances in technology presents many possibilities to support learning in these contexts but schooling and teaching are bemused by the ways they are used for such purpose. To meet this challenge, this paper proposes a personalized approach to learning with the aid of technology. The result is a Personal Learning Environment or a PLE. PLEs are systems that help learners take control of and manage their own learning. It emphasizes among others creation of work products as compared to completion, portfolios instead of grades and a network of learners unlike classmates. The PLE presented in this research is implemented as software running on a popular, highly accessible medium, the world wide web. The mechanism that runs it is based on an internet technology called cloud computing, a model that lessens the complexity of managing and developing web-based software. The key to cloud computing is for an entity such as software to transfer the management of other the layers that make It run to a third party service so it can focus more on its primary objective. In this research external cloud services such as Google Applications, Twitter and others were used to develop a Personal Learning Environment on the web.

1. Introduction

Technology and its advancements have revolutionized the way we live as we knew it; from the way we communicate, the way we socialize, the way we do business and many other things. But schooling and teaching have made little headway even amidst technological developments over the years. Most of the present applications for learning merely reflect traditional paradigms; digital versions of real-world experiences. But even with all the advancements in technology that has given education a new platform in online learning, nothing much has changed about the way learn. The model is still the same."As it was ten years ago, the model is that of a group of people starting at the same time, studying the same materials at the same pace, and ending at the same time" (Downes 2008).

However, it is not uncommon nowadays for students to hold meetings online through chatting. Microblogging allows them to share links to resources. Wikis make them take notes and Wikipedia has proven to be a great source for information (just about anything). Social bookmarking encourages discussions about the contents found in web pages. These tools make teaching and schooling somewhat a bane to young people because they (who are) are more inclined to make use of Web 2.0 software to share, communicate and acquire knowledge. But educational systems and

institutions are, at the very least, bemused at the changing ways in which people use technology to share, communicate and to some extent acquire knowledge.

There's also the issue about how there is little, if not anything at all, in place in terms of software that can help us learn or acquire new competencies after our formal education. But while some of us tend to spend occasional periods in formal training during this time, without a doubt most of the learning we do after this time is informal meaning unaccredited, peripheral, problem-based, learner-driven or simply motivated by interest. In such contexts, formal pedagogy are impractical, even expensive, and while schools have always hummed to a mantra of lifelong learning they have not done a lot to provide for it (Atwell 2007).

A personal learning environment (PLE) has the potential to meet these challenges. It is a highly dynamic, educational, social networking application that provides

an environment where learners get to shape their own learning space. By shaping,

this means cherry-picking learning objects (i.e. reading materials, images, videos) from the web which a learner finds interesting. Then, the platform assesses those chosen objects to form a suitable "curriculum" for the learner. Students can then use their favorite tools to help them organize, process and assimilate information. The PLE is a highly collaborative and interactive platform. It allows for integrating of bits of information across users.

To date, there is no paradigm implementation of a PLE (Lubensky 2006). A lot of conceptual frameworks have been drafted about it. Different individuals have proposed building a PLE out of personal portals such as IGoogle, Pageflakes or Netvibes. There are also ad hoc PLEs in the form of blogs mashed with widgets or plugins. Though learning can be done in these highly personal spaces, they present distractions because they were not meant to be solely for learning and they are not exactly the right fit for the job. Furthermore, information within a web top or a customized blogging platform is not inter operable and manipulable. Hence, a ground up implementation of a personal environment exclusively for learning becomes interesting.

Just as technology has reshaped for the better the different aspects of our lives, so should it do the same with how we acquire knowledge. We need new approaches to learning. We need to consider informal learning which accounts for 80\% of all the learning we do in life (Atwell 2007).

According to the points just mentioned, the paper presents the design and implementation of a cloud based personal learning environment (PLE).

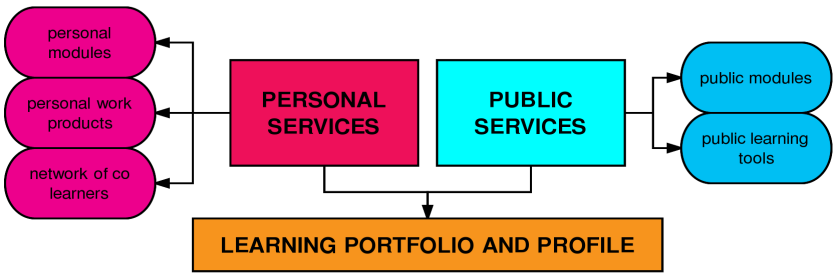
2. The Personal Learning Environment

2.1 Definition

According to Harmelen, Personal Learning Environments are systems that help learners take control of and manage their own learning (Harmelen).

A PLE focuses more on the individual learner's needs. Its learning pedagogy is informal meaning unlike traditional e-learning tools, it is not institutionally centered. One significant difference is that there is no concept of a teacher in a PLE. This means that a learner does not depend on the teacher for lessons, evaluation, and methods of learning. H/she can choose what h/she wants to learn, how h/she wants to learn and when h/she's done, showcase his/her learning and have it assessed. This is because a PLE does not support the traditional classroom style of learning but rather other models such as informal learning, mentoring or on the spot learning, project oriented, inquiry and problem based or learning in a community of practice (Schneider et. al).

The following section and Figure below outlines the underlying concepts in personalized learning as compared to traditional learning (Downes 2006).



2.2 Concepts

2.2.1 Creation and Communication vs Consumption and Completion

The main focus of educational institutions today are the delivery of materials to classes or groups and the fulfilment of course work. Instead of only focusing on the model of consumption and completion, the personal learning environment will be more about creation and communication.

A learner in a PLE should be able to conference with services and co learners. It should have interfaces for creating and manipulating content. It should also have collaboration features that enables a learner to form sets of connections with other learners based on learning objectives. This allows for discussion (for example in forums or instant messaging), collaboration (for instance in the form of writing a wiki) and collective learning action (by coming up with output such as making an open source software or by simply publishing a paper).

2.2.2 Learning Resources

Learning technologies today like Blackboard, Desire2learn or Open Courseware are “based on a publication model of storage and distribution". Their main focus is to deliver materials to classes or groups. But a Personal Learning Environment will not be only be based on this ``principle of access to resources''. A learner in a PLE should also be able to conference with services but also online services that connect students with games, simulations, activities, ad hoc communities of learners, experts and practitioners.

2.2.3 Network of Co-Learners

Traditional learning is like a boxed set given to a group of individuals. The package comes with a set of learning objectives that individuals should meet. To achieve these objectives, everyone uses the same methods. The other people in the group are one's classmates. They will be same people one will be with throughout the course.

Personalized learning is more loose. Instead of groups there is a network. Individuals pursue their own learning objectives. How they pursue these is up to them. They can initiate and sustain a dialogue with others pursuing similar objectives or team up with them. In teaming up there is no leader, instead learners participate according to their interests drifting in an out of discussions as they please

2.2.4 Learning Portfolio and Profile

In a PLE, the products of a learner's engagement and interaction become incorporated in a personal profile. As a way tot measure learning, they are as concrete as test scores and grades. This is because they are a result of different and complex interaction processes. The fact that at the end of the learning process something concrete can be shown allows not only to measure learning but to recognize it as well. "As it becomes easier to simply see what a a student can accomplish, the idea of a coarse-grained proxy, such as grades will fade to the background"(Downes 2008).

2.2.5 Connectivism

In the future, educational institutions will unlikely remain the sole locus of student learning. Rather, they will be part of a much larger learning environment where their offerings are learning objects that will interact with other elements in the ecosystem.

Educational technologists will not only build systems that contribute to a network of resources but also build systems that draw from resources built by others to "create value-added resources" (Downes 2008). This model of learning is called Connectivism and it describes the learning process in networks. In this model, knowledge is not merely contained in the content and creations of educators and learners but more importantly it is the way these contents and creations link together.

This is possible through the use of ontologies. Ontologies are formal representations that describe knowledge. They are useful for applications that search and manipulate information from diverse communities such as a PLE.

3. Hardwire

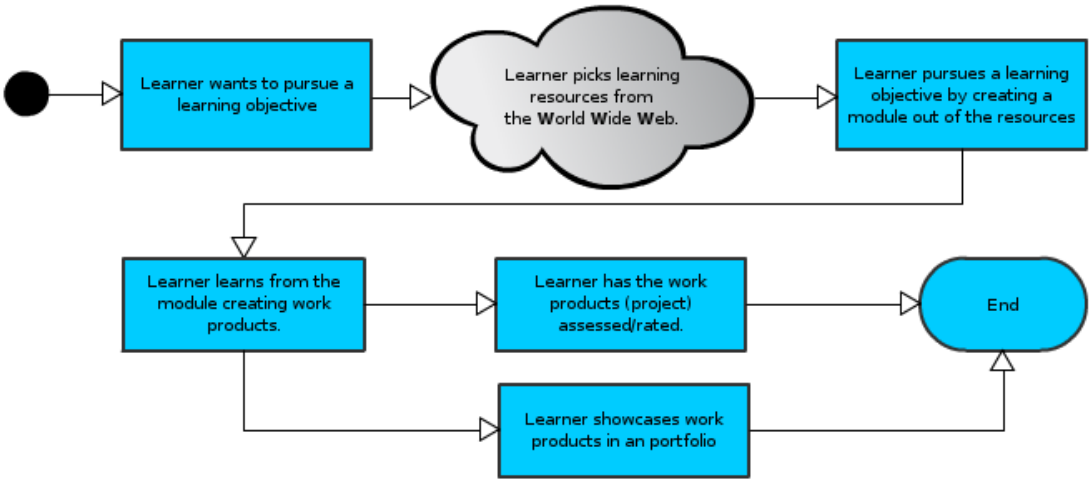
The proponents of this research developed Hardwire, cloud based personal learning environment. As a PLE, Hardwire implements the concepts underlying personalized learning. It has functions for managing content, learning resources, a network of learners and a portfolio.

3.1 Workflow

In Hardwire, the learner, not the teacher or the school, is at the center of the whole learning process. The learning process in Hardwire works with the following units of learning:

- Learning objective -Any fact, technique, behaviour or outcome that a learner wants to achieve at the end of the learning process. In a personalized learning setting, this is problem based or driven by interest.
- Learning resource - Artifacts that are designed to educate learners.
- Work product - Artifacts that show proofs of learning.
- Module - A learning artifact composed of a learning objective with learning resources that will aid the learner in pursuing the learning objective. The analog in traditional learning is a curriculum.
- Project - An assignment of a learner to him/herself showing proofs of his own learning in response to a module. Simply put, it is composed of one or more work products.

A learner can use Hardwire to pursue a learning objective. H/She first looks for learning resources. Traditionally in school they are the books or handouts provided by the teacher.Learning resources and learning objectives are compiled into a module. Learning takes place after. H/She can use any method that suits him/her to pursue the learning objectives h/she set. When h/she is done, h/she comes up with a work product.

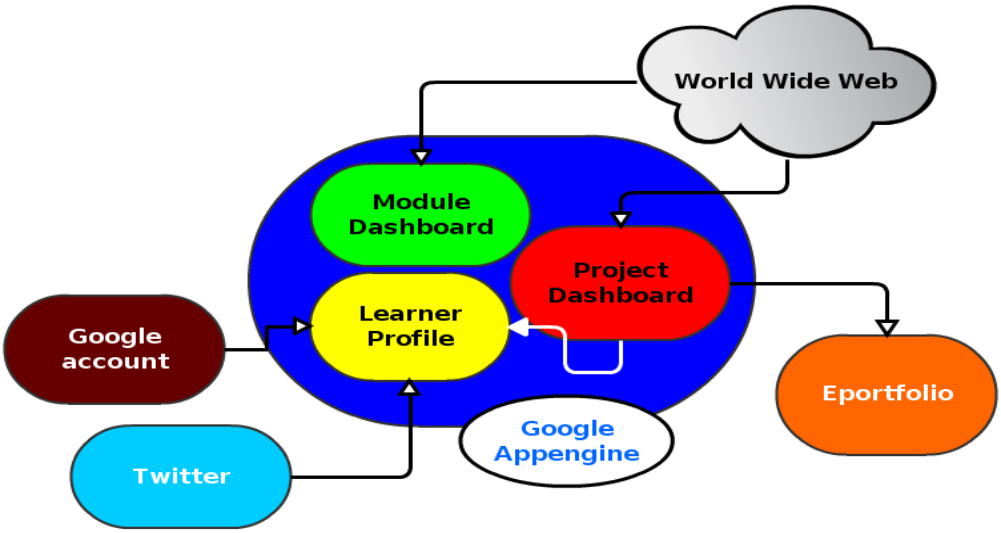


Finally, h/she can have the work product assessed or showcased in a portfolio.

4.2 Conceptual Implementation

The design of Hardwire is shown in the figure below . As one can see from the figure, the Hardwire PLE features a “Module Dashboard" which was built from the

groundup as well as features which are semi mashups of other services such as the ``Project Dashboard'' and the "User Dashboard".



The design of Hardwire

4.2.1 Module Dashboard to organize learning resources

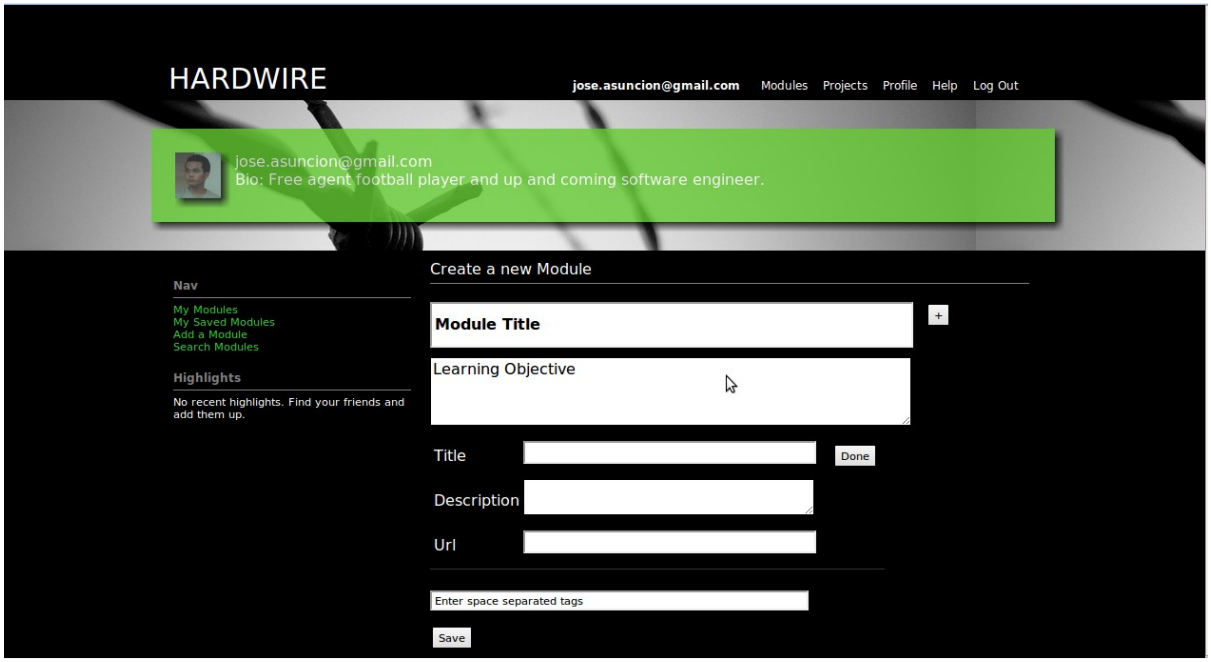
It was previously mentioned that a learner in a PLE should be able to conference with learning resources such as games, simulations and activities. But while it was also noted that traditional educational institutions do not do this, it is unlikely that they will disappear but rather they will evolve to be the providers of these learning resources. Today however, "institutions do not know how to deliver information to other systems beyond interlibrary loans" (Downes 2008).

Instead, learners in Hardwire can view the whole web as their source for learning materials. Each blog, static web page, video, forum thread, instant message conversation etc are potential learning artifacts. The Module Dashboard provides a way for them to come together under one hub. A learner can organize different learning resources under one module. H/She can can give meaning to a module by tagging it. Modules can be shared. There is a search feature that allows other learners to use a module created by another learner for their use

or for the learner to be discovered for possible future collaboration.

4.2.2 Project Dashboard to manage work products

It was also mentioned earlier that a PLE should be more about creation rather than completion of work. While the application does not have specific tools for learning that will enable a learner to create and manipulate content, Hardwire provides a feature for a learner in an online environment to organize and manage his/her own learning. Learners can upload their work product(s), organized under a project and tagged.



Hardwire's Create a Module feature allows a learner to store learning resources related to a learning objective.

Similar to the Module Dashboard,the Project Dashboard's main concentration is to bring learner's work products into a project and have it assessed and less on the the implementation of interfaces for creation and communication. Such will be the focus of future research and development.

4.2.3 Project Dashboard to manage work products

It was also mentioned earlier that a PLE should be more about creation rather than completion of work. While the application does not have specific tools for learning that will enable a learner to create and manipulate content, Hardwire provides a feature for a learner in an online environment to organize and manage his/her own learning. Learners can upload their work product(s), organized under a project and tagged.

Similar to the Module Dashboard,the Project Dashboard's main concentration is to bring learner's work products into a project and have it assessed and less on the the implementation of interfaces for creation and communication. Such will be the focus of future research and development.

4.2.4 Learning Portfolio

As discussed in section 2, personal learning environments should have a way for learners to showcase their work products or have it assessed.

Among the many elearning applications being developed at the University of the Philippines Diliman, there is also simultaneous research going on about eportfolios assessment management systems. Hardwire has collaborated with the developers and integrated one such system into its workflow.

When a project is finalized, Hardwire gives the learner the option to have it assessed. An interface lists down the work products of a project, its details and then an option for assessment. Assessment is done inside the eportfolio that provides an interface for Hardwire to submit work products for assessment in behalf of the user.

But while Hardwire and the eportfolio are two autonomous systems with different aims, it is important to note than in the context of personal learning environments, the eportfolio is just one of the many services that a learner interacts within a PLE.

Hardwire runs on the cloud and makes use of different cloud services for its other functions.

4.2.5 Learner Profile and Networking

The learner profile is basically a summary of a user as a learner in a personal learning environment. It is composed of details about a learner from a brief description, email address and interests and activities.

To foster collaboration among learners, Hardwire has a basic social network function which has add and remove functions. Co-learners are discoverable when searching a module. This makes building up a network in Hardwire based on interest and not social circles. A learner can then look up a learner's profile where his projects, modules and most used tags are shown.

5. Cloud Computing

Cloud Computing is an internet based technology where shared resources, software and information are available on demand. This is a shift from the paradigm where applications are hosted on site or where an I.T department is dedicated to maintaining applications and the platforms and infrastructure they run on. The key to cloud computing is for an entity to transfer the management and maintenance of infrastructure, platforms and software to a third party or a cloud host so it can focus more on innovating their business.

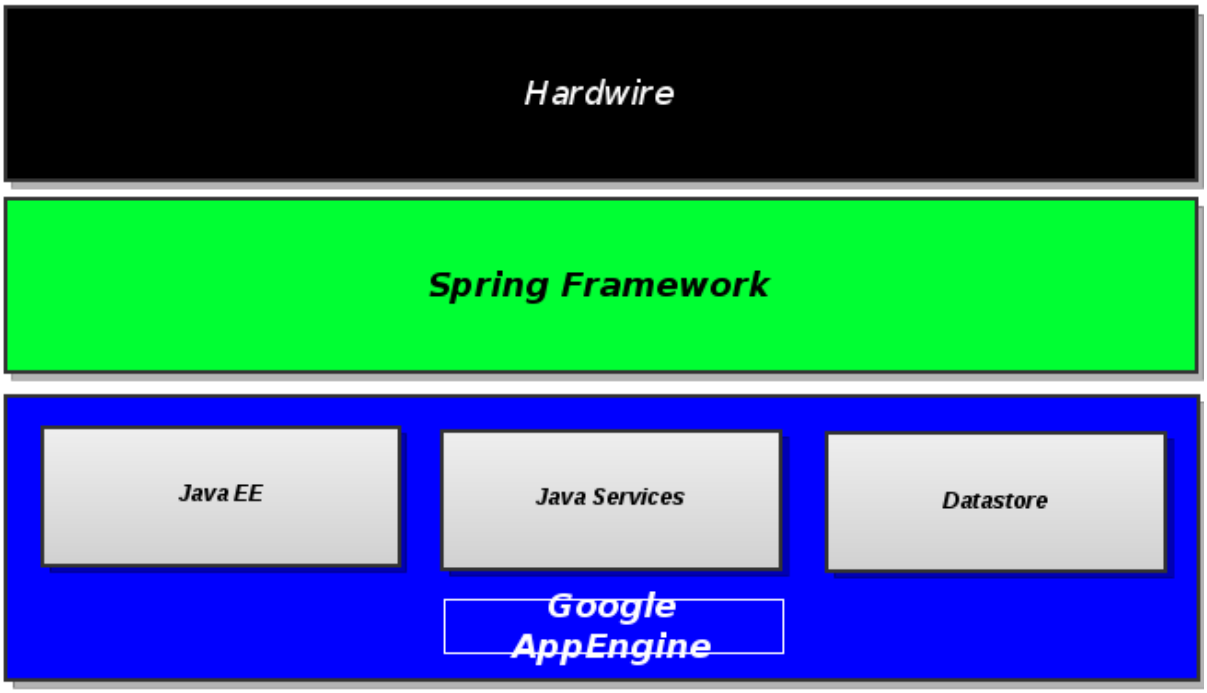
A high degree of scalability and low capital expenditure are just some advantages of hosting on the cloud.

Cloud customers do not have to buy physical infrastructure, purchase expensive software licenses and pay experts to maintain the system. Instead, they only pay for the physical resources (memory, bandwidth, server instances etc) they consume and the applications as use it and when they need it thereby making it cost effective. Gmail is an example of a cloud based application. Compared to Microsot Exchange,

there is no need to buy servers and storage and assemble a technical team to get started with Gmail. Software upgrades isn't a worry either. Google handles all these aspects already.

With a high degree of scalability, the advantages are two-fold. First, a cloud minimizes downtimes related to increasing demand for a hosted application. The resources needed for an application to run are given to it based on how much it needs. When the demand increases the cloud scales almost instantly to accommodate it. This is in contrast to non-cloud applications where resources are pre-allocated. It has the same amount of resources whether demand is high or not. Second, since the application is hosted on a cloud, it is accessible anywhere. Users are not tied to being logged into a private network or being in the office to access applications.

6. Cloud Implementation

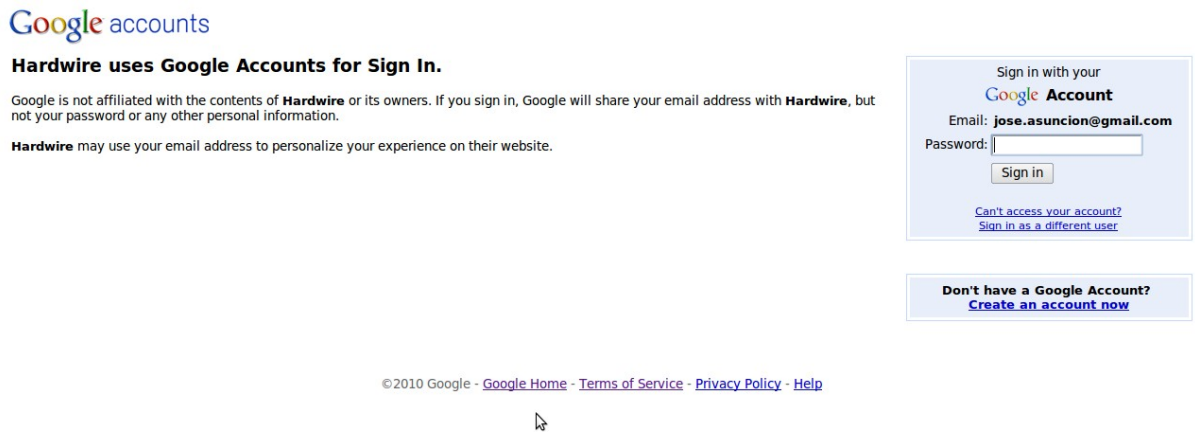


The Hardwire stack featuring Google Appengine as foundation. The Spring framework serves as a container to manage features.

Hardwire runs on Google Appengine's platform. It is composed of the Java Enterprise edition whichmakes it possible to build web applications in Java, Java services composed of commonly used tools in web development and the datastore for storing data. The Spring framework manages the different Hardwire features.

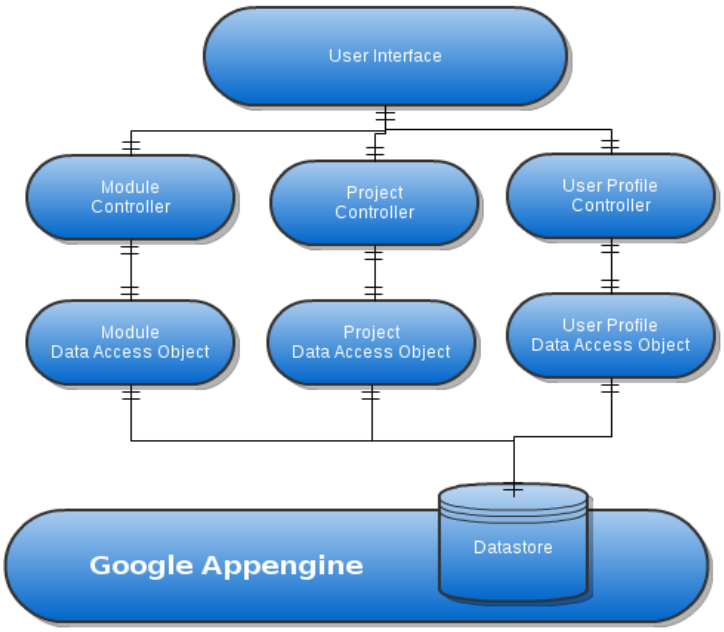
Hardwire is implemented as web based software. All of the services related to a Personal Learning Environment that Hardwire offers were developed from the ground up and deployed on a cloud.

The cloud being used by Hardwire is Google Appengine, a platform that enables developers to build and host web applications in Google-managed data centers. It offers a platform composed of software from runtime environments and a server where the software runs, a database to manage data to hardware layers and delivers them as a service which eliminates and abstracts the cost and complexity of managing them. By doing this, developers only need to focus on writing their applications. There are currently two solution stacks available on Google Appengine, the Python and Java EE platforms. Python and Java are general purpose programming languages with adapters that make it possible to write web applications. Hardwire is written in Java. It use a software framework, Spring, to manage its different features hence adding another software layer Figure shows summarizes the Hardwire stack. Developing on the Appengine however did have its downsides. There were numerous occasions when its datastore went down and new versions of the application could not be deployed. As of the time of this writing, it still a major issue.



Learners can login to Hardwire using their Google account.

Hardwire also consumes services from other web-based, cloud hosted applications such as user accounts and profile information. This benefits the developers of Hardwire in many ways. First, there is no need to develop a complex system to create, maintain, validate and secure user account information. For example, Hardwire makes use of Google accounts to login to the system. Learners in the system are identified through their Google Mail (Gmail) account which is also displayed in their learner's profile. Doing this eliminates the need to build new collaboration features from the ground up but rather it is able to harness the collaboration features already offered by Google in Gmail, Gtalk, Google Buzz and Google Wave. Furthermore,learners can share learning resources through Google Reader. For profile information a function was developed where learners can pull their profile information from Twitter. This information is displayed in the learner profile.



The software architecture of Hardwire shown above follows the Model-View-Controller design pattern, a three layer design that espouses separation of concerns. For example, user interface logic such as actions from a mouse click or button are placed in the User Interface layer show above. Users interact with the user interface and depending on what feature the user is in, the input is handled by a corresponding controller which processes the request and manipulates it. It then accesses the database if it needs more information or wants to return a query by the user. There is a Data Access Object is responsible for managing the database.

8. Conclusion and Recommendation

This paper described a design and implementation of Hardwire, a cloud based personal learning environment.

Through its module dashboard, a learner can pursue his own learning objective whether problem based or interest driven. It also allows a learner to learn the way he wants to learn because h/she can pick his own resources. Furthermore it doesn't limit these resources to materials in school but rather encourages using resources that can be found on the world wide web. The project dashboard provides one way to manage and organize their learning by letting learners upload and tag their work products. When used with the integrated assessment feature, it can give a learner feedback.

Hardwire's features for personal learning were written from the ground up and hosted on a cloud. Furthermore, it also consumes cloud hosted services. The whole Personal Learning Environment that Hardwire offers is thus, cloud-based.

To measure the viability of Hardwire as a learning platform, a form of pedagogical testing should be conducted.

The next phase of development will be to improve the existing features that the implemented PLE has. A learner in a PLE should be able to conference with services but also online services that connect students with games, simulations, activities, ad hoc communities of learners, experts and practitioners. Interfaces

should be made with other learning tools so that learners can create their own content.

Future work will revolve around the implementation of a Proficiency Definition that will serve to guide learners in forming their learning objectives. A recommender system should also be looked into that suggests relevant learning resources. An ontology should will also be applied to describe the modules and projects which are created in Hardwire.

9. References

Downes, S. Nov 2008. The Future of Elearning: Ten years on. http://halfanhour.blogspot.com/2008/11/future-of-online-learning-ten-years-on_16.html. (Accessed March 2010).

G. Atwell. January 2007. PLE - the future of elearning. Elearning Papers. www.elearningeuropa.info/files/media/media11561.pdf. (Accessed June 2010)

R. Lubensky. 2006. The present and future of personal learning environments (ple). blog post, 2006. Deliberations blog. http://www.deliberations.com.au/2006/12/ present-and-future-of-personal-learning.htm. (Accessed March 2010).

M. van Harmelen. Personal learning environments. Wikipedia. http://en.wikipedia.org/wiki/History_of_ personal_learning_environments. (Accessed March 2010)

Schneider, Daniel et. al. "Personal Learning Environments" Edutech Wiki. http://edutechwiki.unige.ch/en/Personal_learning_environment. (Accessed March 2010)

The benefits of the web ontology language in web applications. InDelv.com. http://semantic-web.indelv.com/ the-benefits-of-the- web-ontology-language- in-web-applications.html. (Accessed March 2010)