Towards Developing Installable e-Learning Objects utilizing the Emerging Technologies in Calm Computing and Ubiquitous Learning

Jinan Fiaidhi
Department of Computer Science, Lakehead University
955 Oliver Road, Thunder Bay, Ontario P7B 5E1, Canada
jfiaidhi@lakeheadu.ca

Abstract

*Ubiquitous learning and Calm Computing are hot research topics affecting the field of education for an information-rich world. Calm computing aims to reduce the "excitement" of information overload by letting the learner select what information is at the center of their attention and what information need to be at the peripheral. The objective of calm computing as a new delivery of education is to move e-learning and ubiquitous learning a step further from learning at anytime anywhere to be at the right time and right place with right learning resources and right learning functionalities and collaborative peers. In this paper we report on the adaptation of calm computing technologies in a ubiquitous educational setting with emphasis on the need to cater the preferences of the individual learner to respond to the challenge of providing truly- learner-centered, accessible, personalized and powered with the capability of collaborative learning within a social networking environment. Central to this vision is the development of Installable e-Learning Objects (**Ie_LO**) where learners are able to create, personalize or deploy them over the internet but they do not require the presence of the web browser.*

Keywords: *Installable Internet Applications, Learning Objects, Calm Computing, Ubiquitous Learning*

1. Introduction

The need for computing in support of education continues to escalate. Until recently, everyone assumed that educational computing required desktop computers. Today wireless-enabled laptops, PDAs, iPads and Smart phones make it possible for students to use their time more efficiently, access databases and information from the Internet, and work collaboratively. Connectivity for these devices will soon be the norm rather than the exception. As they become more functional and more connected, the possibility for completely new and unforeseen application increases. Through this flexible learning approach, students can succeed in selectively incorporating critical input from their peers and instructor, then revising their documents based on their own interpretation of facts and theory. We cannot rely anymore on the legacy Web-based distant learning software (e.g. Blackboard, WebCT, WebFuse, CoSE, TopClass, WebEx, VNC, SCORM, and Tango) as they luck portability, ubiquity and scalability as well as they do not support collaborative composition of new learning materials. Actually, the emerging paradigm of Web 2.0 is transforming traditional Internet from a mass media to a social media mode. In Web 2.0 paradigm, applications are delivered through Web browsers and they are responsive to users through rich application interfaces, often including pre-built application services or widgets. Mashups are the essence of such applications. A mashup is an application that combines content from more than one source into an integrated experience. Today mashups are very common and many of the new educational authoring tools are being developed that will produce new applications and services without much programming! However, how to mash up information effectively is a challenging issue. For this reason, the 2008 Horizon report [1] classify mashups into three categories with increasing level of semantics awareness. Figure 1 illustrates these categories.

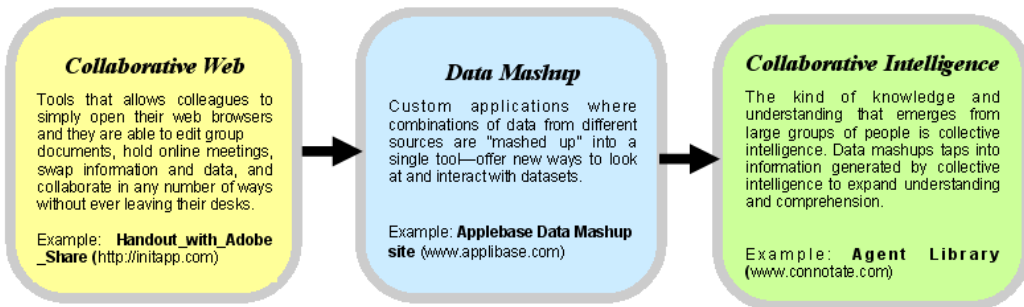

Figure 1. Mashups Semantic Awareness Categories.

Whatever the mashup category is, it generally boils down to the use of some sort of content aggregation technology. 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 or Web Services. Lightweight protocols for content aggregation such as RSS 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, the users find participating in mash-up process as time-consuming. To address this problem, we require an environment that seamlessly integrates devices and services into the physical world as well as to provide semantics for mashing up. Actually, Mark Weiser, a researcher at Xerox PARC, called such environment as "Calm Computing" [2,3,4]. Calm computing enables new ways of processing, integrating and consuming information. In particular, calm technology advocates peripheral awareness of activity in a virtual world and the ability to move easily from a service at the periphery of our attention, to the center, and back according to the learning requests and needs. In this article, we are presenting a general framework that enables the development of calm oriented learning objects that can be developed using any server-side or client-side web technologies and can be installed as a hosted learning application. We call such learning objects as Installable e-Learning Objects (IeLO).

2. Developing Learning Objects: Past and Present

There are many historical attempts to develop a working model for learning objects since R.W. Gerard described this notion in 1967 [5]. Learning objects offer a new conceptualization of the learning process: rather than the traditional "several hour chunk", they provide smaller, self-contained, re-usable units of learning [6]. Learning object design raises issues of portability, and of the object's relation to a broader learning management system. There are mix of many different types of methods, standards and technologies for defining its contents. Figure 2 illustrates the different generations of learning objects environments.

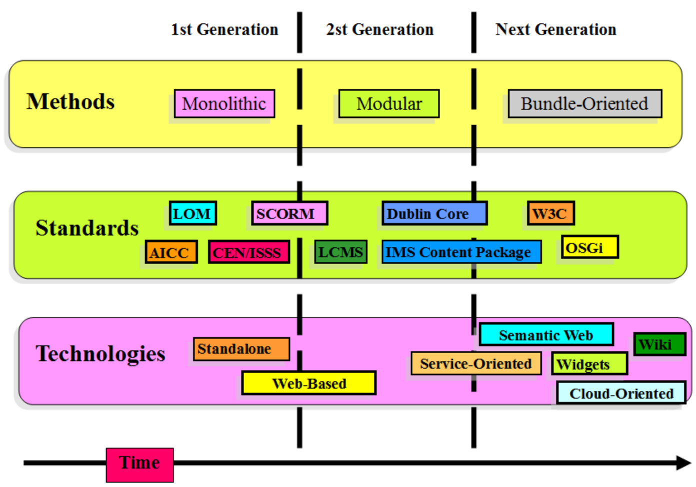

Figure 2. Learning Objects Environments.

Central to any type of learning object is the notion of metadata. Like many other digital objects, learning objects have structures filling with content components, such as learning objectives, procedures, concepts, practice, and assessment. The metadata is used to describe who the creators are, what they are about, and who has what right over them so users can discover, locate, and use these learning objects. Part of the mandate in developing flexible learning objects is to describe the learning object semantics in some form of a schema [7]. Figure 3 illustrates the major components of typical learning objects.

.

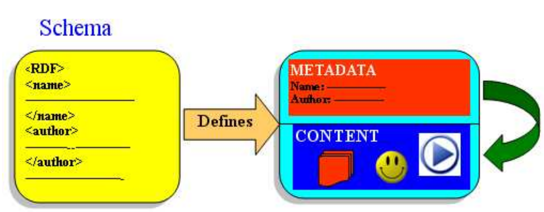

Figure 3. The Essential Components of a Typical Learning Object.

The issues of vocabulary and structures in learning objects let many organizations involved in educational standards like the IEEE Learning Object Metadata (LOM), the Gateway to Educational Materials (GEM) and the Open Archive Initiative (OAI) to develop learning content management systems (LCMS) where learning objects are deposited at a central repository for possible future searching and usage. There are many LCMS notable repositories like eduSource (http://edusource.netera.ca) and MERLOT (http://www.merlot.org/merlot/index.htm). Figure 4 illustrate the structure of a typical LCMS.

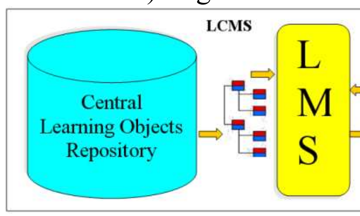

Figure 4. Components of a Typical Learning Content Management System.

To make reusable and interoperable learning objects over the web, we will need to add semantic labels to their components as well as to have technologies to publish them and to enable other user to subscribe to use them or search for their availability, so that application programs can use them for learning purposes. Examples on such web-based systems includes LOP2P [8], LORNet (www.lornet.ca) and LORIS [9]. Figure 5 illustrates the new notion of sharing learning object over the web.

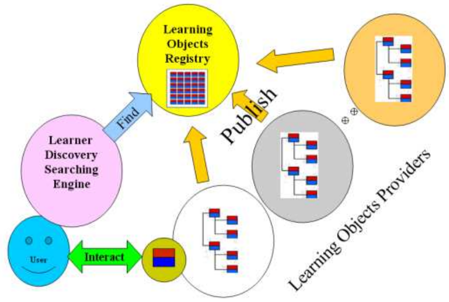

Figure 5. Web-Based Learning Objects.

However, no existing web-based learning object system does promote learning objects mashups. Such functionalities are part of the new technology trend which is generally termed as 'Community Computing' [10]. In fact, the idea of community computing is not totally new. By several projects such as PICO [11] and GAIA [12], community concept had been introduced. Yet despite these interests and the availability of new infrastructures for collaboration and wide accessibility (e.g. Web 2.0 and Cloud Computing) there are very few attempts to redefine learning objects suitable for learning on such new paradigms.

3. Reshaping Learning Objects for Next Generations of Learning Systems

Technology is rapidly changing and we are all excited on what will e-learning look like in a few years time? When Stephen Downes laid down his manifesto for e-Learning 2.0

in 2005, he tapped into the zeitgeist of emerging social technologies and theorized a number of possibilities. After six years on using this technology a reappraisal of learning within such ever evolving digital spaces is overdue. There are many drivers that may change or reshape learning including: Distributed and Cloud computing, Smart mobile technology, Collaborative intelligent filtering and Multimedia visualization, integration and mashing-up technologies. At the same time there are many learning applications that utilize some of the mentioned driver's technologies (e.g. MUPPLE [13], CME [14], Jampots [15]). In such evolving arena several researchers tried to envision the new shape of learning objects that rise to the expectation of the new learners utilizing the emerging digital technologies such as mashups (e.g. [16], [17]). Certainly mashups represent a hallmark of the new Web 2.0 trend technology. In a recent article [18], the Widget notion has been proposed for reshaping the structure of the new learning objects. Web widgets represent portable chunks of code that can be installed and executed within any HTML-based web page easily by learners and hence they present themselves as new way for knowledge aggregation. Widgets looks and acts like traditional apps but they are implemented using web technologies like JavaScript, Flash, HTML and CSS. Running any Widget requires the activation of the web browser. Figure 6 illustrates this vision.

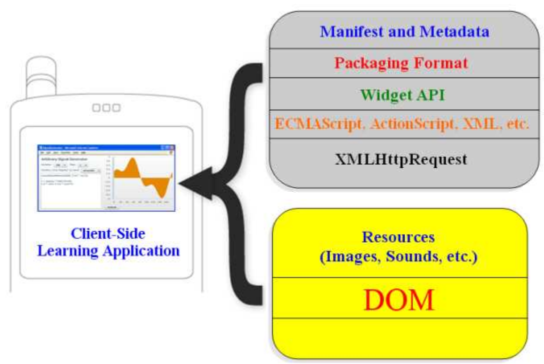

Figure 6: Widget-Based Learning Objects.

However, there are many problems associated with adopting Widgets for learning objects besides their dependancy on the Web browser. Here are some of these problems:

- Not all browsers support scripts, therefore, users might experience errors if no alternatives have been provided
- Different browsers and browser versions support scripts differently, thus more quality assurance testing is required
- More development time and effort might be required (if the scripts are not already available through other resources)
- Developers have more control over the look and behavior of their Web widgets; however, usability problems can arise if a Web widget looks like a standard control but behaves differently or vice-versa
- There are many Widget standards (e.g. W3C Widget 1.0, Widgetbox, Spring Widget, Yahoo Widgets, Microsoft Windows Live Widgets and Google Gadgets) and languages.

4. Towars Installable Internet Learning Applications for Calm Computing

In order to overcome the Widget limitations, we need to adopt a new technology that enables knowledge aggregation and be resident on the client device as well as be able to be accessed over the internet without the need for the browser activation. In this direction, the Apache Pivot provides an innovative solution through its Installable Internet Application (IIA) technology [19]. The technology is an open-source platform for building Rich Internet Applications (RIA) (http://pivot.apache.org/). With the Apache Pivot one can eaisly create web-based learning applications without the need for the web browser as Pivot develops such applications in Java and hence it can be executed by any JVM compliant platform. Indeed, the Pivox allows also the development of applications using more generic language like the BXML language (i.e. an XML compliant) which can be serialized and processed by any other platform. For these reasons, Pivot provided a viable option for developers who want to build rich client applications in Java (or any other JVM-compatible language). If we compare Pivot application development to althernative technologies such as Adobe Flex, Microsoft Silverlight and Oracle JavaFX, we can easily favor Pivot. Actually, the Adobe Flex applications are written in ActionScript, Silverlight applications are built using .NET, and JavaFX applications are built with JavaFX Script. Pivot allows Java developers to build rich internet applications using technologies and APIs developers already know without the need for the browser. Pivot offers a comprehensive set of standard UI elements and several features that simplify the development of modern GUI web applications. These include data binding, animated effects and transitions, web services integration, and an XML markup language called WTKX for declaring the structure of a UI. We are going to call the Pivot IIA learning applications as Installable e_Learning Objects (Ie_LO). However, the use of such applications over the internet without the assistance of the web browser for the purpose of wider ubiquitous learning requires further support from the following technologies:

- **Publish-Subscribe Technology:** A lightweight technology for publishing Ie_LOs over the internet and to allow other learners to discover it and syndicate their feeds.
- **Mashup technology:** To allow Ie_LOs to be mashed and assmbled to form larger learning applications.
- **Calm Technology:** To enable learners to personalize their learning objects and harvest Ie_LOs over the internet according to their context. The harvesting technology need to use some sort of semantics for searching relevant Ie_LOs. The harvested Ie_LOs stays at the center of the user attention while the not relevant Ie_LOs needs to be collected by the Ie_LOs garbage collector to return to the Ie_LOs periphery registry.

Although there are many publish-subscribe technologies, the OSGi bundle model would work very well with the type of Ie_LO applications. With OSGi, applications can easily be managed remotely because remote management is inside OSGi's genes. This always has made OSGi easy to use in clusters and various other distributed environments like the cloud computing. Moreover, the advantages of the OSGi service model are even more effective. A web based collaborative environment is by definition a dynamic environment. Adding instances, removing instances, and instances that fail will likely influence the other instances. This means that the application will need to handle the dynamicity of the services that these computing instances provide. There will be also dependencies that must be managed. OSGi services shine in these areas, making it relatively simple to correctly model these dynamic dependencies. So overall the combination of the Pivot Le_LO applications and OSGi is clearly an interesting one and OSGi will provide several essential services such as the identification of Ie_LO by wrapping them as a bundle and centeral registery for future discovery. Based on this combination we can distinguish between the Le_LO residing at the center and those at the periphery for calm computing purposes. The learner can process the Pivot bundle at the client side (i.e. the center of the user attention) by running the OSGi Equinox container while he can invoke remote bundles through the OSGi Felix container at a remote OSGi server like the Apache CXF. However, the most important functionality in any calm computing learning application is the ability of harvesting relevant learning objects from the periphery repository and moves it to the learner center of attention. For this purpose we need to have an underlying model of the knowledge for describing learning objects. Regular, or syntactic, learning objects have structured metadata text with some untyped hyperlinks. Semantic-based learning objects, on the other hand, provide the ability to identify information about the data within learning objects, and the relationships between learning objects, in ways that can be queried or exported like a database. Imagine a semantic-based learning system devoted to food. A learning object for an apple would contain, in addition to standard text information, some machine-readable semantic data. The most basic kind of data would be that an apple is a kind of fruit. The learning system would thus be able to automatically generate a list of fruits, simply by listing all learning objects that are tagged as being of type "fruit." Further semantic tags in the "apple" learning object could indicate other data about apples, including their possible colors and sizes, nutritional information and serving suggestions, and so on. These tags could be derived from the learning object metadata text but with some chance of error - accordingly they should be presented alongside that data to be easily corrected. If the learning system periphery exports all this data in RDF or a similar format, it can then be queried in a similar way to a database - so that an external learner could, for instance, request a list of all fruits that are red and can be baked in a pie. However, to implement new learning applications (e.g. food cataloguing system) with semantic capabilities requires a lot of functionality dealing specifically with ontologies and metadata. Currently, needed functionalities are typically created for each learning application individually, requiring a lot of work, time and specific skills. Being able to lower these implementation costs would be hugely beneficial. In developing any learning ontological system, finding and selecting the right concepts and instances is a central task of its own in ontological user interfaces. For end-user applications, any search usually begins by first finding the right concepts with which to do the actual ontological querying. For efficient semantic content indexing, accurate indexing entities need to be found with as little effort as possible. Also ontology developers need concept search when creating links between concepts, especially when developing distinct, yet heavily interlinked ontologies. For this purpose, finding and harvesting relevant learning object from the periphery requires sort of semantic wiki. Actually, wikis replace older knowledge management tools, semantic wikis try to serve similar functions: to allow users to make their internal knowledge more explicit and more formal, so that the information in a wiki can be searched in better ways than just with keywords, offering queries similar to structural databases. The amount of formalization and the way the semantic information is made explicit vary. Existing systems range from primarily content-oriented (like Semantic MediaWiki) where semantics are entered by creating annotated hyperlinks, via approaches mixing content and semantics in plain text, via content-oriented with a strong formal background (like KiWi), to systems where the formal knowledge is the primary interest (like Metaweb), where semantics are entered into explicit fields for that purpose. Also, semantic wiki systems differ in the level of ontology support they offer. While most systems can export their data as RDF, some even support various levels of ontology reasoning. Indeed the use of Semantic MediaWiki may help in maturing knowledge about learning objects and hence developing various mashups. However, for achieving mashups between OSGi Ie_LO and other browser based learning contents one needs a mashup server that collaborated with browser like the Apache XAP.

The final service required for learning in the context of calm computing is the ability to take learning objects out of the user center of attention when it is not any more required by his current status of learning. This process is traditionally called garbage collection. Garbage collection systems were first developed around 1960 and have undergone much research and refinement since then. The mechanism of garbage collection is fairly simple to describe although the implementation is more complicated. The garbage collector's goal is to form a set of reachable objects that constitute the "valid" objects in your application. When a collection is initiated, the collector initializes the set with all known root objects such as stack-allocated and global variables. The collector then recursively follows strong references from these objects to other objects, and adds these to the set. All objects that are not reachable through a chain of strong references to objects in the root set are designated as “garbage”. At the end of the collection sequence, the garbage objects are finalized and immediately afterwards the memory they occupy is recovered. There are several points to note if we want to implement a garbage collector for Ie_LOs:

- **The collector needs to be conservative.** It never compact the OSGi container by moving blocks of memory and updating pointers. Once allocated, an object always stays at its original memory location.
- **The collector needs to be both request and demand driven**. The implementation makes requests at appropriate times. You can also programmatically request consideration of a garbage collection cycle, and if a memory threshold has been exceeded a collection is run automatically.
- **The collector needs to run their own thread.** At no time all threads can be stopped for a collection cycle, and each thread is stopped for as short a time as is possible. It is possible for threads requesting collector actions to block during a critical section on the collector thread's part.
- **The collector needs to use hybrid strategies (Open and Closed).** Most garbage collection systems are “closed”—that is, the language, compiler, and runtime and thus the collector needs to collaborate to be able to identify the location of every pointer reference to a collectable block of memory. In contrast to closed collection systems, “open” systems allow pointers to garbage collected blocks to reside anywhere, and in particular where pointers reside in stack frames as local variables. Such garbage collectors are deemed "conservative." Their design point is often that since programmers can spread pointers to any and all kinds of memory, then all memory must be scanned to determine unreachable (garbage) blocks. This leads to frequent long collection times to minimize memory use. Memory collection is instead often delayed, leading to large memory use which, if it induces paging, can lead to very long pauses. As a result, conservative garbage collection schemes are not widely used. However, it is possible to strikes a balance between being “closed” and “open” by knowing exactly where pointers to scanned blocks are wherever it can, by easily tracking "external" references, and being "conservative" only where it must. By tracking the allocation age of blocks, the collector implements partial (“incremental”) collections which scan an even smaller amount of the heap. This eliminates the need for the collector to have to scan all of memory seeking global references and provides a significant performance advantage over traditional conservative collectors.

Figure 7 illustrates the overal architecture of the proposed calm computing based learning system.

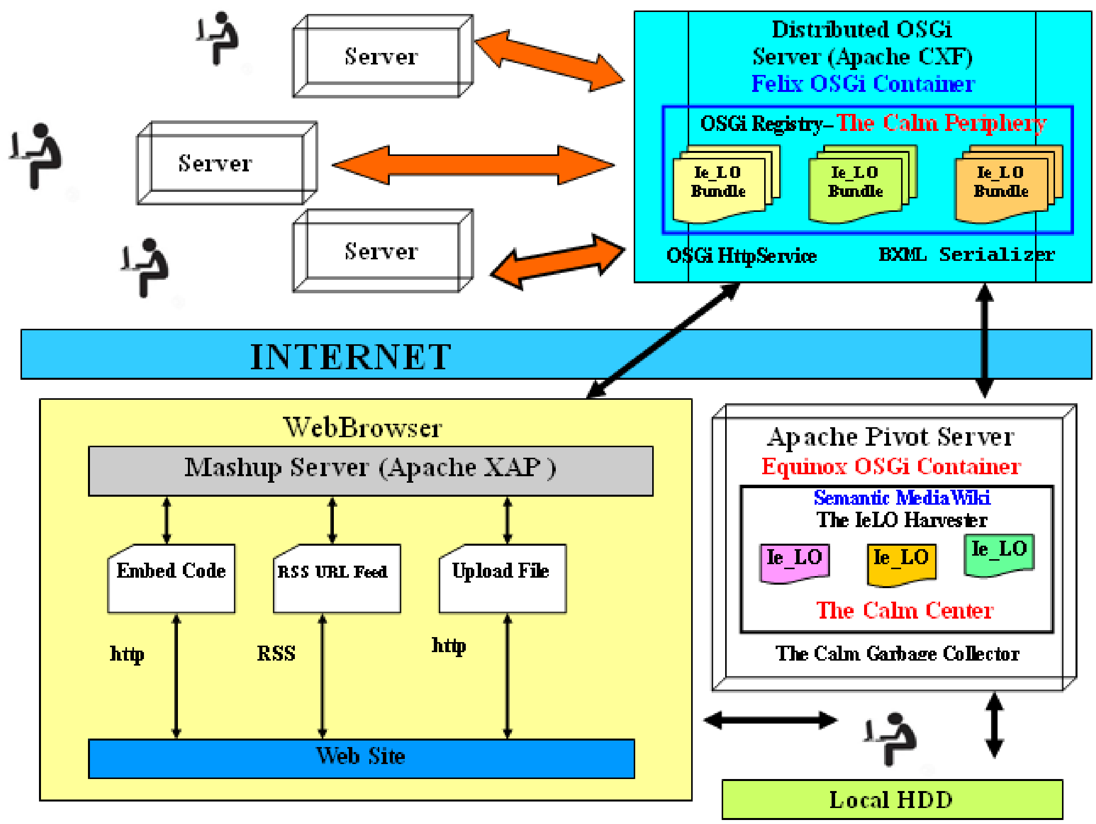

Figure 7: An Overall Architecture for Sharing Ie_LOs for Calm Computing.

5. Conclusions

This article describes our vision to the future generations of learning systems that are based on calm computing technology. Central to this vision is the representation of learning objects as lightweight Pivot bundles where the learner can publish them and search for others for further knowledge maturing purposes. This article also illustrates how the notion of learning objects has been evolved during the last three decades. This article also introduced a new notion for learning objects based on the new web-based calm computing. The new notion is called Installable e-Learning Objects (Ie_LO). Two major calm computing services are described in some detailed in our vision: The harvesting of Ie_LO from the repositories of learning objects and the Ie_LO garbage collection for the learner’s centeral memory of learning objects. The research described in this visionary article is far from complete as it currently works in progress. A comprehensive prototype is almost complete that utilizes a cloud computing infrastructure [20].

Acknowledgment

This article is an enhancement to my tutorial presentation of the FGIT conference [18]. Also I would like to thank NSERC for supporting my research.

References

[1] The Horizon Report, The New Media Consortium and EDUCAUSE, ISBN 0-9765087-6-1, www.nmc.org/pdf/2008-Horizon-Report.pdf, 2008.
[2] A. Tugui, “Calm Technologies in a Multimedia World,” ACM Ubiquity, vol. 5, no. 4, 2004.
[3] M. Weiser, “The Computer for the 21st Century,” Scientific American, pp. 94–100; Sept. 1991. www.ubiq.com/hypertext/weiser/SciAmDraft3.html.
[4] Jinan Fiaidhi, Wes Chou and Joseph Williams, Mobile Computing in the Context of Calm Technology, IEEE IT-PRO, Editorial Article, May-June 2010.
[5] R. W. Gerard, Shaping the Mind: Computers In Education. In R. C. Atkinson & H. A. Wilson, Computer-Assisted Instruction: A Book of Readings. New York: Academic Press, 1969.
[6] Beck, Robert J., "What Are Learning Objects?", Learning Objects, Center for Int. Education, University of Wisconsin-Milwaukee, http://www4.uwm.edu/cie/learning_objects.cfm?gid=56 , 2001.
[7] Ben Daniel, Honggang Wu, "Developing a Schema for Learning Object Based on Object Oriented Model of Object Inheritance," icalt, pp.439, Third IEEE International Conference on Advanced Learning Technologies (ICALT'03), 2003.
[8] Rafael de Santiago and Andre´ Raabe, Architecture for Learning Objects Sharing among Learning Institutions—LOP2P, IEEE Transactions on Learning Technologies, 3(2), APRIL-JUNE p91-95, 2010.
[9] Simone L. de Moura, Fábio J. Coutinho, Sean W. M. Siqueira and Rubens N. Melo, Integrating Repositories of Learning Objects Using Web-Services and Ontologies, International Journal of Web Services Practices, Vol.1, No.1-2 , pp. 57-72, 2005.
[10] Y. Jung, J. Lee and Minkoo Kim, Community Computing Model Supporting Community Situation Based Strict Cooperation and Conflict Resolution, In Software Technologies for Embedded and Ubiquitous Systems, Lecture Notes in Computer Science, Volume 4761, 47-56, 2007.
[11] M. Kumar, et al., PICO: A Middleware framework for Pervasive Computing, In Pervasive Computing 1268-1536, pp72-79, 2003.
[12] R. Jennings, Developing Multiagent Systems: The Gaia Methodology, ACM Transactions on Software Engineering and Methodology, 12(3), pp317-370, 2003.
[13] F. Wild, F. Mödritscher and S. Sigurdarson, Designing for Change: Mash-Up Personal Learning Environments, eLearning Journal, www.elearningpapers.eu, Nº 9, ISSN 1887-1542, July 2008 .
[14] M. Al-Zoube, Int. Arab Journal of e-Technology, Vol. 1, No. 2, PP58-64, June 2009.
[15] Bo Dong et. al, “Jampots: A Mushup System Towards an E-Learning Echosystem”, IEEE Fifth International Conference on INC, IMS and IDC, 2009.
[16] B. Taraaghi, M.Ebner and S. Schaffert, Personal Learning Environments for Higher Education: A Mashup Based Widget Concept, Proceedings of the 2nd Int. Workshop on Mashup Personal Learning Environments (MUPPLE09), Nice, France, 2002.
[17] Malinka Ivanova & Tatyana Ivanova, Involving students in managing their own learning, eLearning Journal, www.elearningpapers.eu , Nº 21, ISSN 1887-1542, September 2010.
[18] Jinan Fiaidhi, Developing Personal Learning Environments Based on Calm Technologies, In T.-h. KIM et. al (Eds),Proceedings of the UNESST 2010, CCIS 124, PP134-146, Springer-Verlag, 2010.
[19] Greg Brown, Introducing Apache Pivot, Technical Report, Apache Pivot, Sept. 29, 2010.
http://svn.apache.org/repos/asf/pivot/project/presentations/Introducing%20Apache%20Pivot%202010-09-29.pdf
[20] Sabah Mohammed, Daniel servo, and Jinan Fiaidhi, HCX: A Distributed OSGi Based Web Interaction System for Sharing Health Records in the Cloud", 2010 International Workshop on Intelligent Web Interaction (IWI'10), Toronto, Canada, August 31, 2010.

Author

Dr. Jinan Fiaidhi is a full Professor and the Graduate Coordinator with the Department of Computer Science, Lakehead University, Ontario, Canada since late 2001. She is also an Adjunct Research Professor with the University of Western Ontario. She received her graduate degrees in Computer Science from Essex University (PgD 1983) and Brunel University (PhD, 1986). During the period (1986-2001), Dr. Fiaidhi served many academic positions at many institutions (e.g. University of Technology (Asso. Prof and Chairperson), Philadelphia University (Asso. Prof), Applied Science University (Professor), Sultan Qaboos University (Asso. Prof.). Dr. Fiaidhi research is focused on mobile and ubiquitous learning utilizing the emerging technologies (e.g. Cloud Computing, Calm Computing, Mobile Learning, Personal Learning Environment, Enterprise Mashups, Semantic Web). Dr. Fiaidhi research is supported by the major research granting associations in Canada (e.g. NSERC, CFI). Moreover, Dr. Fiaidhi is a Professional Software Engineer of Ontario (PEng), Senior Member of IEEE, member of the British Computer Society (MBCS) and member of the Canadian Information Society (CIPS) holding the designate of ISP. Dr. Fiaidhi has intensive editorial experience (e.g. Editor of IEEE IT-Pro, Associate EiC of the Journal of Emerging Technologies in Web Intelligence). More information on her publications and news can be found at her Web page: http://flash.lakeheadu.ca/~jfiaidhi