A Data Mashup for Dynamic Composition of Adaptive Courses

Mohammed Al-Zoube and Baha Khasawneh
Princess Sumaya University for Technology, Jordan

Abstract: *This paper presents a novel adaptive course composition system that based on mashing up learning content in a web application. The system includes three major components, static course system, data mashup, and adaptive course composer. The first system enables lecturers to build multiple lesson courses that are managed with corresponding APIs including documents stored in Google docs or videos located at YouTube. The data mashup utilizes some web and search engines APIs in which learners can search for relevant web content and display it in a grid layout. The adaptive course composer uses learning object from the data mashup to build adaptive course based on learner preferences. It uses a selected number of the IEEE LOM to sort the learning objects according to their educational role, difficulty level, or rating, and hence, eliminate the use of course domain ontology.*

Keywords: *Data mashups, adaptive course composition, Google apps, learning objects.*

Received November 16, 2008; accepted December 1, 2008

1. Introduction

The next generation of internet tools and technologies, often referred as Web2.0, are making it much easier for learners to become more active participants in their own education. These emerging technologies highlight the new challenges facing educational institutes in a fundamental way, thus influencing their approach in teaching and the process of learning. The Horizon 2008 report [8] recognized six areas that are of great impending to influence the way academic teaching and learning for the next five years. These challenges include the use of learning videos, collaboration webs, mobile broadband, data mashups, collective intelligence, and social operating systems. Some of these areas, such as collective intelligence and social operating systems, are uncommon in the academic world but is used in industry and entertainment which opens the way for academia to adopt such technologies to advance creative thinking and learning.

While such emerging technologies indicate a significant shift in the way we view information technology and communication, it has created a greater need for a leadership that embraces change and innovation at all levels of educational institutes. This includes not only faculty, but also students and administration who could positively influence the use of information in many forms. It is imperative that the academic community would adopt such technologies and practices in experimentation to promote and demonstrate its value by taking advantage of opportunities for interdisciplinary work and collaboration.

Personalized eLearning shapes the learning experience to each individual learner. It employs an active learning strategy which allows the learner to be in control of the context, pace and scope of his learning experience. It supports the learner by providing tools and mechanisms through which they can personalize their learning experience, and hence, can help to improve learner satisfaction with the learning experience. However, the design of adaptive educational systems requires significant effort [4], since dependencies between educational characteristics of learning resources and learners characteristics are too complex to figure out all possible combinations.

This paper presents the development of a novel web-based adaptive course composer system based on mashing up learning contents. Data mashups, which defined by the Horizon report as “a web application that combines data from more than one source via a single, unified tool” have great potential in educational environments. Mashups also are playing a significant role in development of personal learning networks. Individuals are able to use mashup applications to combine various web based resources together into their own custom learning environment. The rest of this paper organized as follows. Section 2 gives an overview of data mashups. Section 3 briefly describes Google applications and services used to build our application. The proposed application is introduced in Section 4. Finally, Section 5 ends this paper with conclusion and future work.

2. Data Mashups

The ever increasing amount of free web content (text and audiovisual), that is available for internet users is unprecedented. However, the needs of non-technical end-users are not well served due to the lack of integration of such content or its presentation in a way that would not always serve their needs. A natural approach to solve this problem is the integration of such content and available services that might be presented in way which best serve users needs' by using mashups. Web services are programs interface that uses the web as a mean to provide services by the use of APIs with semantics instead of only HTML.

Data mashups is combining accessible web-based content and services in order to create new applications and services that would support end-users needs and offer better view of the data from different sources. While all needed content and services to accomplish the integration is already available, mashups provide tools that can help users in creating new applications and services but requires programming expertise in many areas including HTML, scripting and database. Such expertise would create systems that include content extraction from different web pages, filtering or adding metadata, integrate it with other local or remote DB sources, and ultimately direct the output to a variety of services, such as map services.

According to Programmableweb.com, a mashup community, more than a thousand mashups and some mashup programs or tools that help users in building their own mashups are realized over the internet. Tools like Yahoo! Pipes[1] and Google Mashup Editor[2] provide users with the ability to create applications, filter, and combine content from the Web and pull updates whenever available. Mashup Scripting Language (WMSL) [11] is a new programming model for the integration of multiple web services into a single page and allows users to use scripts in order to combine HTML and create mapping relations to achieve this integration. Some visual programming tools such as Data mashups[3] and Anthracite[4] allow users with little programming background to create a website with common functions found on a web application and also to include some data-flow mechanism. C3W [7] allows users to create a spreadsheet like application and to utilize certain functions from different web applications and to display the gathered content on a compelling way based on users' needs. Internet Scrapbook [12] is an interactive tool that enables users to extract components of different pages and bring them on one web page. More common use of data mashups is using Google maps (Gmap) combined with other available services such as housingmaps.com, ProgrammableWeb.com, and ikozma[5] to name few.

Mashups in education is still lacking behind despite its obvious potentials as envisaged in the Horizon 2008 report. Recently, many universities from all cotenants are converting their mail and calendaring systems to a single system powered by Google. For example, Arizona State University (ASU) approved and forged a partnership with Google to implement Google apps for education [1] and to migrate from IMAP email client to Gmail. This cooperation would enable ASUs' academic community, mainly students and faculty members, to create a personal web page that includes, Gmail, and Gmaps with the ability to integrate additional services and application. A regular student page would evolve to look like an Amazon like page that includes course suggestions and similarity of activities and other services such as online courses, news headlines, weather forecasts, and other calendar related features [13].

However, Google Apps for education are not truly full mashup applications rather a standalone application that can utilize Google services combined. Branzburg [2], argued and suggested the use of Google mashups in k-12 education. Such as the inclusion of Gmaps Pedometer in physical education, Google plain meter in math and geography, and YourGmap for elementary students to locate and mark some neighbourhood locations such as schools or even playing pizza houses.

3. Google Apps

Google Apps is a collection of web-based programs and data storage that run in a web browser, without requiring users to buy or install software. Users can simply log in to the service to access their files and the tools to manipulate them. The offerings include communication tools (Gmail, Google Talk, and Google Calendar), productivity tools (Google Docs: text files, spreadsheets, and presentations), a customizable start page (iGoogle), and Google Sites (to develop web pages). Google Apps allows institutions to use their own domain name with the service and to customize the interface to reflect the branding of that institution. In this way, a college or university can offer the functionality of Google Apps in a package (and with a URL) that is familiar and comfortable to constituents.

Several colleges are switching from more traditional applications to the single Google system with minimal hardware or software requirements for the improved performance and security measures. For example, Macalister switched from Oracle collaboration suite to Google application for education since Oracle was not stable enough according to their

[1] http:// pipes.yahoo.com
[2] http:// code.google.co/gme
[3] http://www.datamashups.com
[4] http://www.metafy.com/products/anthracite
[5] http:// www.ikozma.com

assessment [9]. Furthermore, Google's mission to organize the world's information and make it universally accessible and useful requires making it available in contexts other than a web browser. For this purpose, Google provides APIs to let client software request information outside of a browser context. In the following subsections, a brief description is given for some of the APIs that have been used in our system.

3.1. Google Data

Google Data (GData[6]) is a new protocol that uses the syndication mechanism to send queries and receive query results. It is based on the Atom 1.0 and RSS 2.0 syndication formats, plus the Atom Publishing Protocol. Syndication is an effective and popular method for providing and aggregating content. GData provides a general model for feeds, queries, and results to send queries and updates to any service that has a GData interface. GData also lets users send data and update data that Google already have. Several Google services provide a Google Data API, including Google Base, Blogger, Google Calendar, Google Spreadsheets and Picasa Web Albums.

3.2. Google Docs

Google Docs[7] is a web-based word processing application that allows users to create, view and share documents, presentations, and spreadsheets over the Internet. Each file has a creator, who determines who is allowed to access the file, either as a viewer (with read-only rights) or a collaborator (who can change the file). Because Google stores all of the files and content centrally, collaboration and document management become far simpler than when distributing files to multiple people and keeping track of different versions. For a text document, for example, the application maintains the file, allowing authorized users to see or edit the text while keeping track of all the changes and who makes them. The Google documents list data API allows client applications to view and search through documents stored in Google Docs using Google Data API feeds. The client application can request a list of a user's word processing documents and upload existing documents to Google Docs.

3.3. YouTube

YouTube[8] is a website where users can upload and share videos. It contains millions videos tagged by users and categorized into predefined set of categories such as education, entertainment, comedy, etc., One of an emerging class of social applications, YouTube allows users to post videos, watch those posted by others, post comments in a threaded-discussion format, search for content by keyword or category, and create and participate in topical groups. YouTube ties into several blogging applications, giving users a quick way to blog about a particular video and include a link to it. It also offers an open access to key parts of the YouTube video repository and user community, via an open API interface and RSS feeds.

Using YouTube APIs, client application can easily integrate online videos from YouTube's rapidly growing repository of videos. Client application can request a list of videos that match specific search criteria, upload new videos, edit video metadata such as comments and favourites, and modify user settings. YouTube is increasingly being used by educators as a pedagogic resource. Many educators believe that the act of creating content is a valuable learning exercise, helping develop a deeper understanding of the subject matter and the tools used to create that content. It has the potential to expose students to new insights and skills, as well as link them to various online communities.

3.4. AJAX Search API

The Google AJAX Search API[9] provides simple web objects that perform inline searches over a number of Google services (web search, local search, video search, blog search, news search, book search, and image search). In the context of our application, the AJAX search service is used to provide the data mashup with web search results.

3.5. Google Base

Google Base (GBase[10]) can be seen as self describing semi-structured database where users can upload their structured data. Google base adopts a very simple data model in which users describe their data using an item type and attribute value pairs without any restriction on names and values. In our context, Google base is used to store the values of learning objects. With Google Base Data API, users can query Google Base Data to create applications and mashups, as well as input and manage Google Base items programmatically. An application can upload new data, update or delete existing items, and execute specialized queries to find matches for complex attribute criteria.

3.6. Code Search

Google Code Search[11] helps to find function definitions and sample code by providing one place to search publicly accessible source code hosted on the

[6] http://code.google.com/apis/gdata
[7] http://code.google.com/apis/documents/
[8] http://code.google.com/apis/youtube/
[9] http://code.google.com/apis/ajaxsearch/
[10] http://code.google.com/apis/base/
[11] http://www.google.com/codesearch

Internet. In addition, users can write regular expressions to search more precisely, or restrict the search by language, license or filename. Moreover, one can view the source file with links back to the entire package and the webpage where it came from.

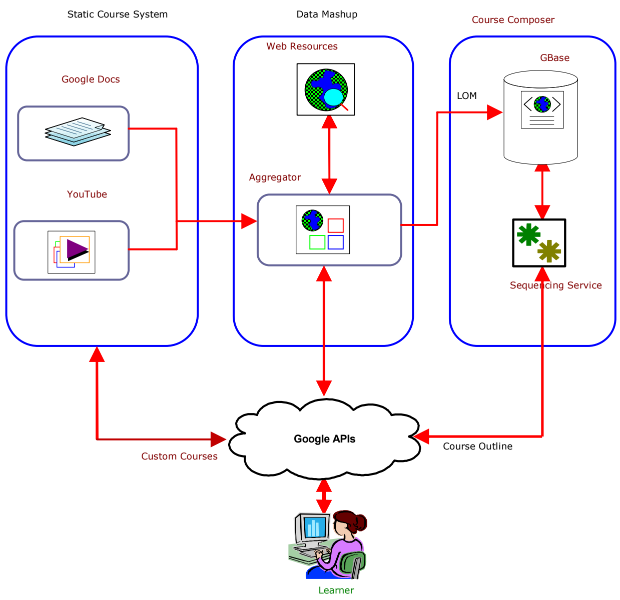

Figure 1. Architecture of the system.

4. Architecture of the Proposed System

The main architectural components of the proposed dynamic course composition system are shown in Figure 1. The first component is the static course system which provides courses with predefined materials consisting of documents, presentations, and videos. The second component is the data mashup which collect information from the web using several searcher components, and the third component is the dynamic course composer. In the following subsections we provide more details about the functionality of each component.

4.1. The Static Course System

The main objective of the static course system is to provide students with custom courses with predefined materials. Each course consists of lessons that are designed by lecturers according to the course subject graph. The course contents could be documents, presentations, or videos, which are hosted in Google Docs and YouTube applications. The lessons in each course provide essential knowledge every learner should acquire that enables him/her to build upon advanced or new knowledge from the evolving web.

The Docs API also enables full-text search over the documents, hence, it is more convenient for students to search for specific information in the lessons. Figure 2 shows an example of some of the courses and a number of lessons within each of the course. For example, Web Programming course contains the following lessons: Basic HTML Elements, CSS Styling, Page Layout, DHTML, and Image Arrays that could be viewed or searched from within the same page. The lessons provided by the static course system represent the basic knowledge every learner should acquire in order to build upon any further knowledge.

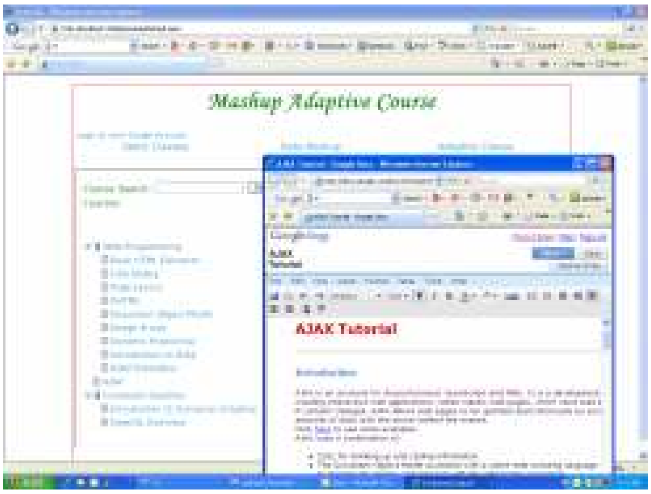

Figure 2. Static courses search and listings.

4.2. The Data Mashup

The main motivation for building the data mashup is to provide learners with the latest research results, books, publications, tutorials from the dynamic web. Indeed, students spend long time on the web searching and browsing for information to gain knowledge and skills. The web has several search engines that are designed to search for specific kind of data from specific resources. For example, Google has search engine for images, videos, code, etc., and from specific resources like YouTube, blogs, and maps.

The data mashup component uses APIs of several web and data sources search engines to display the retrieved results in grid layout. In our system the following APIs were utilized: Google AJAX search API (with web, video, blog, and book searchers), YouTube API, Google code API, and Google Docs API. This component plays a major role in providing students with a better and more convenient visualization of retrieved search results. For example students can search Google for a specific or a broad area of information such as multimedia or image processing, and the results would be displayed in a fashion as shown in Figure 3.

The search results in the data mashup may contain rich media resources, called Learning Objects (LOs). These LOs can be reused for other purposes, such as learning object recommendation and personalized learning path generation, if described appropriately with sufficient metadata. Several standards exist to describe learning resources, among them are the

Dublin Core Metadata (DCMI)[12] initiative and the IEEE Learning Object Metadata (LOM) [10]. The IEEE LOM is a major part of the Shareable Content Object Reference Model (SCORM) [6], proposed by Advance Distributed Learning (ADL) organization to enable interoperability between content and learning management systems. IEEE LOM provides an information model that defines the structure of a metadata instance for a learning object. IEEE LOM contains about 70 elements grouped into nine categories: general, lifecycle, meta-metadata, educational, technical, educational rights, relation, annotation, and classification.

Figure 3. Search results for HTML in the data mashup.

When a user finds a web resource (YouTube video, for example) with a good educational value, she can recommend this learning object to the admin of the system. The admin checks the contents and the quality of the learning object and decides on its suitability as a learning object. A candidate learning object could be annotated with suitable metadata and saved for further uses. The following metadata elements of learning objects are adopted from the IEEE LOM:

- URI: a global unique identifier for the learning object [1.1 general].
- Title: name given for the learning object [1.2 general].
- Topic: a keyword describing the topic of this LO [1.5 general].
- Difficulty: how hard is to work this LO [5.8 education].
- Date: date this LOM is created [8.2 annotation]
- Educational role: learning resource type to represent the potential role for the learning object [5.2 education].
- Concept: added as an additional element to describe the concept covered by this LO.
- Prerequisite concept: added as an additional element to describe the prerequisite concept of this LO.
- Rate: added as an additional element to rate the LO.

Learning objects metadata are stored in GBase which is a metadata management service that hosts and provides access to the learning object metadata associated with learning objects.

4.3. The Adaptive Course Composer

Adaptive eLearning offers the potential to uniquely address the specific learning goals, prior knowledge, and context of a learner so as to improve that learner's satisfaction with the course. Authoring such adaptive courses has typically been a very complex, time consuming and expensive task [3, 4]. Intelligent Tutoring Systems have failed to be adopted as a mainstream approach to personalized eLearning in higher education or secondary/tertiary education due their inflexibility and composition costs. The composition of an adaptive course requires input from various modelled entities such as the learner, the concept space, the pedagogical strategy(s), the content and the adaptive mechanisms influence the composition and realization of an adaptive course.

To provide a simple mechanism for building adaptive courses, a system that utilizes the learning objects collected from the data mashup to generate adaptive course outline for a specific topic dynamically is proposed. This outline provides a guide to the learner about what needs to be learned and in what order. Adaptive course construction method should support the course composer in identifying what parts of the course need to be adapted, and what criteria should be used for this adaptivity [5]. For example the course composer should be able to specify that the course be adaptable based on the learners' prior knowledge, or on the learners' preferred communication or collaboration style.

A software component that sorts the learning objects respecting the concept sequence and the educational role sequence is built. The algorithm allows the user to specify parameters such as difficulty level and educational role of the retrieved learning objects. Usually, a course should cover a series of concepts with specific predefined relationships between these concepts, e.g., the prerequisite relationship. The following steps are followed to form the course outline:

1. The user enters a topic of the course, and optionally the difficulty level, the educational role (e.g., introduction, definition, explanation, example), or top rated LOs.
2. A query is built to search GBase for learning objects using GBase API.
3. The retrieved leaning objects are grouped and ordered by the position of the educational role of each object.

[12] http://dublincore.org/

4. Within each group, learning objects are sorted according to their difficulty level and rate.
5. For each learning object in the introduction group:
   a. Get the learning object from the remaining groups with the same concept.
   b. If the learning object has a prerequisite concept, place the learning objects of this concept first.
6. Repeat step 5 for the remaining groups.

To illustrate the functionality of the algorithm, assume that a learner needs to query the system about HTML as a general topic. After specifying the topic and the optional parameters, the system uses GBase query API to retrieve a set of learning objects that matches the query. Then, the above algorithm is used to arrange these learning objects and results as a sequence of hyperlinks representing an outline for HTML course as shown in Figure 4. It should be noted that the outline generated from the learning object may not be linear, since the learning objects may not be fine grained as they are authored by different people and collected from the internet. However, this part of the course complements the static one and gives the learner extra material based on learners annotation.

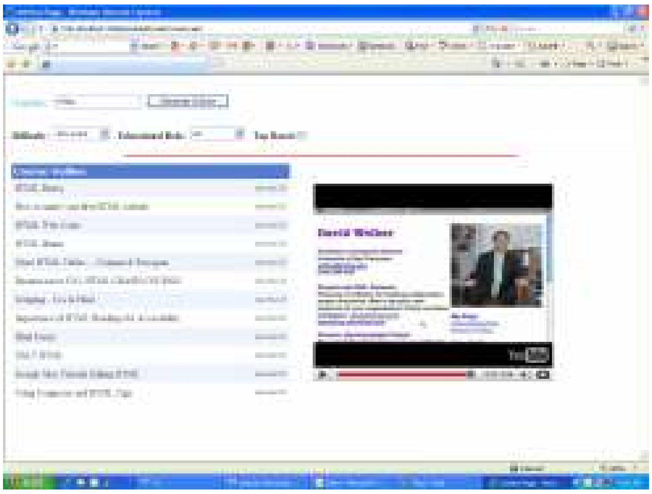

Figure 4. Adaptive course outline generated for HTML and a YouTube lesson.

5. Conclusions

In this paper, we presented a web-based multimedia e-learning system with adaptive courses to improve not only the way custom courses are built but also how the information can be organized, searched, or viewed. This system is designed mainly as an e-learning mashup application to enable educators and learners to design adaptive courses that are made up of a number of lessons stored on Google or YouTube with searchable capabilities, and a dynamically generated lessons based on mashing up information from different web resources. In traditional adaptive elearning systems, the delivery of learning material is personalized according to the learner model, while the materials inside the system are a priori determined by the system designer. However, our system enables both ends of the learning process to make use of available learning objects over the Internet and a proper set of metadata, to build the adaptive course without the need for course or learner ontology. In addition, the system has the following features:

1. Service oriented architecture based on Google apps and services, and hence, could easily be extended.
2. Support for lifelong learning with minimum cost.
3. Uses collaborate knowledge for adaptation.
4. Full-text and tag-based search for learning objects.

Future work on the system includes the implementation of modules for reusing learning objects in hybrid recommendation system.

References

[1] Barlow K. and Lane J. "Like Technology from an Advanced Alien Culture: Google Apps for Education at ASU," *in Proceedings of Special Interest Group on University Computing Centers*, Indiana, pp. 8-10, 2007.

[2] Branzburg J., http://www.tecllearneing.com, 2008.

[3] Chiali S., Eberrichi Z., and Malki M., "A Dynamic Composition of an Adaptive Course," *in Proceedings of World Academy of Science, Engineering and Technology*, USA, pp. 11-11, 2006.

[4] Dagger D., Wade V., and Conlan W., "Personalization for All: Making Adaptive Course Composition Easy," *Computer Journal of Educational Technology and Society*, vol. 8, no. 3, pp. 9-25, 2005.

[5] Farrell R., Liburd S., and Thomas J. "Dynamic Assembly of Learning Objects," *in Proceedings of WWW2004*, USA, pp. 17-22, 2004.

[6] Fletcher W., Handwork M., and Herald S., "Sharable Content Object Reference Model," *Technical Document*, USA, 2004.

[7] Fujima J., "Clip Connect Clone: Combining Application Elements to Build Custom Interface for Information Access," *in Proceedings of User Interface Software and Technology*, USA, pp. 175-184, 2004.

[8] Horizon Report, "The New Media Consortium and the EDUCAUSE Learning Initiative," http://www. nmc.org/ pdf/2008-Horizon-Report. pdf, 2008.

[9] Ledig A., http://www.themacweekly.com, 2008.

[10] LOM IEEE 1484.12.1-2002 http://ltsc .ieee.org/doc/ wg12/ LOM _1484_ 12_1_v1_ Final_ Draft.pdf, 2002.

[11] Sabboh M., "Web Mashup Scripting Language," *WWW2007*, ACM press, pp. 1305-1306, 2007.

[12] Sugiura A. and Koseki V., "Internet Scrapbook: Automating Web Browsing Tasks by

Demonstration," *in Proceedings of User Interface Software and Technology*, USA, pp. 9-18, 1998.

[13] Wong J. and Hong J., "Making Mashups with Marmite: Towards End User Programming for the Web," *in Proceedings of CHI2007*, USA, pp. 1435-1444, 2007.

Mohammed Al-Zoube received his BSc and MSc in electrical engineering from Jordan University of Science and Technology in 1990 and 1994, respectively. In 2002, he received his PhD in computer science from University Science Malaysia, specializing in multimedia systems. Currently, he is working at the Department of Computer Graphics and Animation, Princess Sumaya University for Technology, Jordan.

Baha Khasawneh received his PhD degree in computer science from The George Washington University in 2000. He is currently an assistant professor and chairman of the Department of Computer Graphics and Animation at Princess Sumaya University for Technology. His research is focused on multimedia systems mainly in image processing, multimedia, transmission and protection.