Designing CMS modules to support language learning

Paul Daniels
Kochi University of Technology
daniels@kochi-tech.ac.jp

This article presents an overview of Course Management Systems [CMS] and how a CMS can be modified to support language learning environments. First, the author justifies the advantages of a CMS while examining how CMS fit into the realm of learning theory and pedagogy. Next, several crucial differences between open source and proprietary CMS applications, and the essential strengths and weaknesses that need to be considered when selecting a CMS are outlined. The main part of the article focuses on a particular open source CMS called Moodle and explains how it is possible to modify or add modules or blocks to Moodle in order to supplement or enrich language instruction.

Introduction to CMS

Web-based management systems have become a valuable element of instructional framework design for both language courses and content specific courses. Because of rapid developments in information technology, only recently have we been able to digitalize, process, and distribute vast amounts of information over high-speed networks. In education, particularly second language education, digital information needs to be managed or organized in a simple and logical fashion to ensure learners do not become overwhelmed. By customizing open source CMS, educators can better fabricate learning environments that facilitate the construction of knowledge for today's 'plugged-in' language learners.

CMS are software applications that typically run on a web server or network server and allow educators to easily manage course and student data through a web browser interface. The primary functions available in CMS are to organize and distribute course content, administer learning exercises or quizzes, and track student progress. While CMS software is often used to manage distance learning courses, it is just as popular for distributing supplemental learning content outside of the traditional face-to-face classroom.

CMS software products are widely used by colleges and universities to manage and deliver course content via the Web. For example, MIT uses *Stellar*; the University of Michigan uses *CHEF*; and Stanford uses *CourseWork*. These are systems that the universities have designed for their own use and are not available for public download. Universities without adequate resources to develop their own personalized CMS can download and install an open source CMS such as *Moodle*, *.LRN*, or *Sakai*.

Background

Why use a CMS? While there are just as many arguments for and against using a CMS as there are for and against using technology in general, there are three arguments in favor of adopting a CMS that are worth investigating.

The first argument is that a CMS can facilitate instructor creation of constructivist learning tasks and environments. Before outlining the benefits of CMS for language learning, it is helpful to examine the origin of constructivist learning theory. The work of Lev Vygotsky (1978) and Jean Piaget (1972) paved the way for constructivist approaches to learning, which propose that learners actively construct their own knowledge from resources made available to them. The collaborative nature of today's wired environments hold promise for constructivist learning activities. The benefits of networked collaboration appeared early in the Internet boom when USENET was developed as an open and shared forum where people from around the world collaborated for the purpose of learning. More recently, CMS have appeared in every facet of teaching and learning. When crafting language learning activities for or with a CMS, it is helpful to consider learning activities which are based on cognitive and constructivist science. A useful resource is Jonassen's (2007) taxonomy of meaningful learning, which offers a concrete guideline for developing digital material or activities. According to Jonassen, meaningful learning is that which is active, constructive, intentional, authentic, and cooperative. These assumptions encourage instructors to think about CMS as a tool that students learn with rather than a tool to teach with.

A second argument for incorporating a CMS is enhanced multimedia support. Firmly tied to course design and CMS development should be the role of multimedia, and how multimedia can be better used to personalize and customize learner knowledge and experiences. Mayer (2001) attempts to ground research in the area of multimedia in cognitive theory. Based on numerous cognitive studies, Mayer suggests that learners have two separate channels for processing information, one visual and one verbal, and that these channels can become overextended. He argues that multimedia systems can be more effective for learning since they address both the visual and verbal channels. CMS are designed to streamline the sharing of multimedia instructional material as well as integrate text-based activities together with multimedia. Using a CMS to deliver multimedia enhanced learning content may also help to strengthen leaner motivation. Astleitner and Wiesner's (2004) model of multimedia learning and motivation offers a useful framework to consider when supplementing language learning activities with multimedia elements.

The third argument for the adoption of a CMS relates to modern society's recent technological advancements. It is now clear that the way in which youths communicate and process information is dramatically changing. Educators need take into consideration the rapidly

evolving digital-age teaching and learning environments expected by 'generation Y' learners. Marc Prensky, a futurist in educational technology, has written extensively on 'digital natives' and 'digital immigrants'. Prensky (2001) argues that digital natives are fundamentally different from digital immigrants and proposes that "students' brains have physically changed – and are different from ours – as a result of how they grew up". Because of fundamental differences between digital natives and digital immigrants in the way they think and process information, teachers need to re-develop old content and instructional styles to better suit today's learners. This doesn't mean that teachers have to reinvent the wheel, but changes in instructional methodology are needed. Prensky offers a few recommendations for changes, such as teaching content at a faster pace, reducing the amount of step-by-step instruction, and giving students more choice of learning activities. Deploying a CMS to serve supplemental course material and activities, particularly collaborative activities, multimedia enhanced activities or educational games, would be a dramatic step towards redefining instructional methodology to better match today's learners.

Selecting and implementing CMS

As CMS become more prevalent, educators need a clear understanding of the realities of balancing learning and management. New Web 2.0 tools such as discussion forums, blogs, wikis, and shared whiteboards have made it simpler to create teaching activities that involve collaborative document creation, multimedia publication, and social networking. However, administrators and teachers must also be sure that the CMS they select can support tasks that center on promoting, rather than discouraging, student learning. In addition, repetitive language practice activities, which often are easiest to develop and deploy, must be structured to be meaningful within the context of the overall course.

For educators and administrators, deciding which CMS product is most suitable for an institution can be an intimidating and formidable task. The suitability of a CMS is determined by the effectiveness of its content and how efficiently learners interact with that content. CMS can support the administration and deployment of collaborative learning activities and tasks, with their main strength being the organization and distribution of content, rather than content creation.

Choosing a CMS requires "test driving" the different tools and administrative interfaces of each product to determine whether all necessary tools and functions required for a given learning environment are included. Some products offer a sand-box, an area for testing applications on line, while other CMS products must be downloaded and installed to determine their suitability. For a helpful comparison of CMS applications, EduTools <www.edutools.info> offers a well-organized community supported website. A review of the most recent articles and research on CMS products is also recommended before making a final decision on which system is most suitable for both student learning preferences and instructional methods.

Kochi University of Technology deployed *Moodle* <moodle.org> as the primary CMS for language instruction in 2004. *Moodle* was chosen as it is one of the leading open-source CMS available (Adkins, 2005). The acronym '*Moodle*' stands for 'Modular Object-Oriented Dynamic Learning Environment'. *Moodle* was developed by Martin Dougiamas and its

design is grounded in cognitive and constructivist science in order to provide more effective pedagogical support in different learning environments (Dougiamas & Taylor, 2003). Before deciding to employ *Moodle*, several alternative systems were investigated, including *.LRN* <dotlrn.org>, *Sakai* <sakaiproject.org> and *Webclass* <www.webclass.jp>. In the end, *Moodle* appeared to be the best solution because it is open-source, it is extremely flexible in that it allows us to add and edit modules, and it boasts a large community of users who provide support. An open source alternative appeared to best meet both learner and instructor demands at our language center given the fact that in-house support for administering the system is available and student administration can be accommodated through a single server installation.

Why open source?

Lawrence Lessig (1999), a leading expert on free speech, open source and the right to innovate, provides a straightforward answer; "some architectures invite innovation; others chill it". At Kochi University of Technology, the main motive or adopting an open source CMS was to have the flexibility to edit the code, create new activities, and integrate the system with other applications to complement our current curriculum. While there are many promising CMS applications on the market, some are rather inflexible. A CMS, whether it be open-source or propriety will never possess all of the functions desired, and while *Moodle* has many valuable features, perhaps as many as if not more than proprietary systems, it lacks several features that are important to the success of language learners. Because of *Moodle*'s wide user base, vibrant community of developers, and ease of development, it was possible to add the necessary functions to support the language learning activities unique to our institution. One of the main strengths of *Moodle* is the ability to add modules or block to the core application without actually changing the core code. This is extremely important in a CMS in order to ensure the upgrade process goes smoothly. The following section describes some of the limitations of using out-of-the-box *Moodle* for language learning and how it is possible to add modules or blocks to create exciting new language learning activities.

Course management system module development

During a three year period at Kochi University of Technology, 5 modules were either improved upon or developed from scratch in order to enhance the functionality of the *Moodle* CMS. Most of the modules were designed as *Moodle* plug-in modules or as blocks. The modules were mainly designed using PHP, the scripting language that *Moodle* is written in, and MySQL, a popular SQL database engine supported by *Moodle*. In addition to PHP and MySQL, Adobe Flash was used to enhance the multimedia aspects of *Moodle*.

The *Moodle* modules discussed in the following section are being used to support instruction in several ESP classes, for example, Science Lab, Science English, Science Reading, Technical Writing, and Technical Writing & Presentation courses, at Kochi University of Technology. The modules were installed on the department's onsite course management server running Linux *Redhat Enterprise* and *Moodle* 1.8.4. All of the modules are open source and can be downloaded from <blog.netcourse.org/>.

Improved media support

The *Moodle* blog module was redesigned to allow students to upload a greater variety of media files to a course blog using a mobile device. With the current version, text, images, sound, and video files can be uploaded by mobile phone or via a web browser and displayed on a student's blog site. The original mobile blog module was implemented three years ago and was successfully integrated into the English curriculum with 400 students enrolled in first year English courses. Students are able to upload files to the course system using mobile device such as a mobile phone. Students have since uploaded over 10,000 images along with English descriptions of images depicting their daily lifestyles. The collection of blog images and entries is an example of authentic student-produced content that can be used in future learning activities.

In 2007, the system was improved and several new multimedia plug-ins were added. The new media blog module allows for text and video uploads, and also audio file and video file uploads. Using the audio/video conversion tool, the module now supports thumbnailing of any uploaded video file. Typically, mobile phones can upload video format in 3GP format, which is a multimedia container format used with 3G mobile phones. Since 3GP is a simplified version of the MPEG-4, the video content can easily be played back using a browser plug-in such as *QuickTime*.

In addition to the audio and video upload functions, students can also record and save audio files directly to their course blog using a standard browser and Adobe Flash Player. This new media function allows learners to create voice boards by easily embedding voice interactions into a web page. Using this technology, teachers or learners can set up podcasting activities within the *Moodle* CMS.

The voice recording technology was designed using *Actionscript*, PHP and the Adobe *Flash Media Server*. A simple Adobe Flash object, as seen in the illustration below, is embedded into an html page. Students can record and play back their voices by clicking on the record and play buttons. Once they have a satisfactory recording, they save the file. The audio files are saved as a Flash FLV files using Adobe *Flash Media Server* and can be played from within the CMS as streamed media. Students also have the option of recording their voice on an external recorder and uploading an mp3 file to their blog. In order to use this module, *Flash Media Server* needs to be installed. An Open Source alternative to Adobe *Flash Media Server* for recording and streaming Flash files is Red5. Figures 1 and 2 below show a sample student blog.

Figure 1: Browser-based voice recorder and media file upload field.

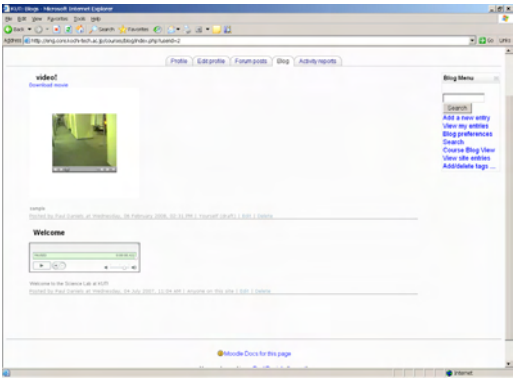

Figure 2: Video and voice blog entries

Presentation module

While many instructors use *PowerPoint* presentations to supplement their lectures, it is not necessarily the best medium for distributing content online. *PowerPoint* files are often several megabytes in size, require proprietary software for viewing, and are not efficient in maintaining the file structure of external sound or video files. The presentation module was designed to automatically convert Microsoft *PowerPoint* files to Adobe Flash format after the files are uploaded to *Moodle*. Users can upload their presentations and add comments to each others' presentations and attach multiple choice questions to their presentations. In addition, teachers can attach existing *Moodle* course quizzes to uploaded presentations. Converting *PowerPoint* to Adobe Flash is beneficial as it allows for more accessible learning content. *PowerPoint* is proprietary software, whereas Adobe Flash player is free and Gnash is a GNU Flash movie player. In addition, the Flash file size is considerably smaller than the *PowerPoint* file making presentations easier to distribute via the web.

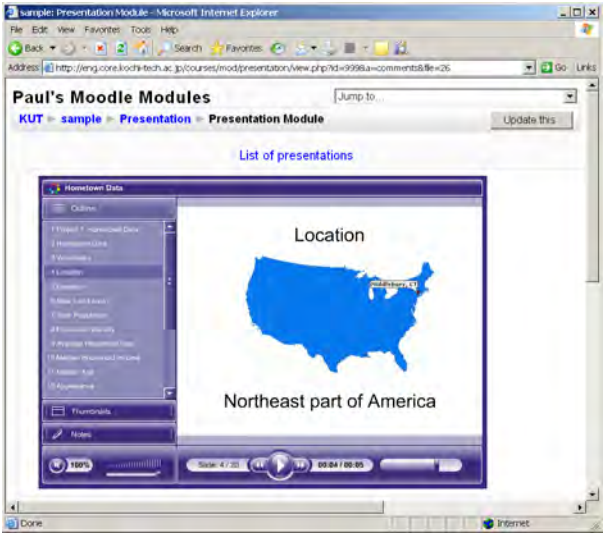

Figure 3: A *PowerPoint* file displayed in an Adobe Flash environment.

Text analysis module

While some learners or educators may be critical of the use of computerized text analysis to assess student writing, recent research supports the validity of computer-scored writing activities (e.g., Laufer & Nation, 1995; Goodfellow, Lamy, & Jones, 2002). More recently, a software application called *e-Rater*, developed by the Educational Testing Service is used to score GMAT essays. The practice of using computers to rate essays is relatively new, but it is a promising approach. Computerized text analysis will not replace human raters; rather it can be used to complement the human rating process.

The text analysis module analyzes student-produced text in a *Moodle* forum, blog, chat, or journal. Student data can easily be collected and analyzed by a single click of a button. Text data, such as total word count, total unique words, number of sentences, words per sentence, hard words, lexical density and a Fog index, can be downloaded as a CSV file and imported in Microsoft *Excel*. This module can be used to assist instructors with the tedious aspects of writing assessment. It can also be incorporated as a self-monitoring task for student writers.

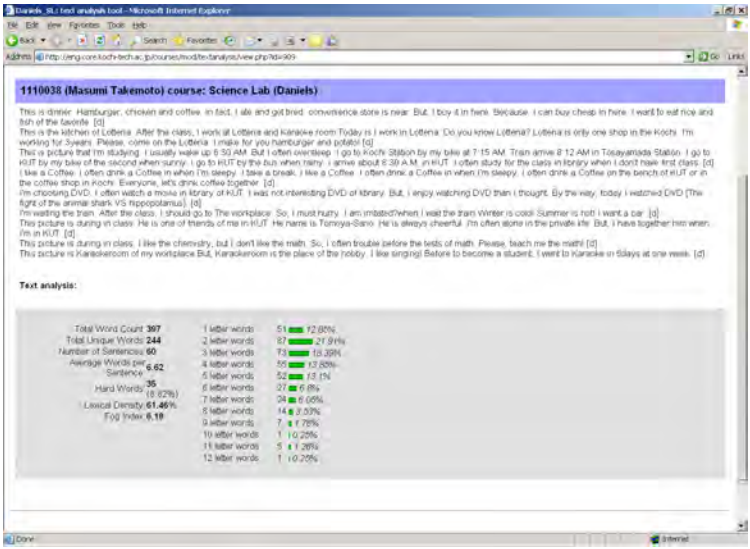

Figure 4: Sample analysis of a first year student's writing

Mail quiz module

Using the email quiz module, students can sign up for a study topic and receive content and quiz questions via a user-defined email schedule. The script also tracks correct/incorrect responses and average quiz scores. This module is currently being used to introduce TOEIC vocabulary to students studying for the TOEIC exam. The study content includes the first 1,000 most frequently used TOEIC words with definitions and examples in both Japanese and English, and multiple choice vocabulary quizzes, all of which is sent out to students' mobile devices.

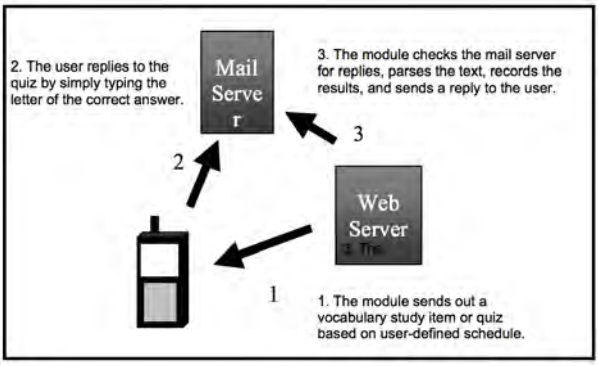

Figure 5: Mail Quiz backend

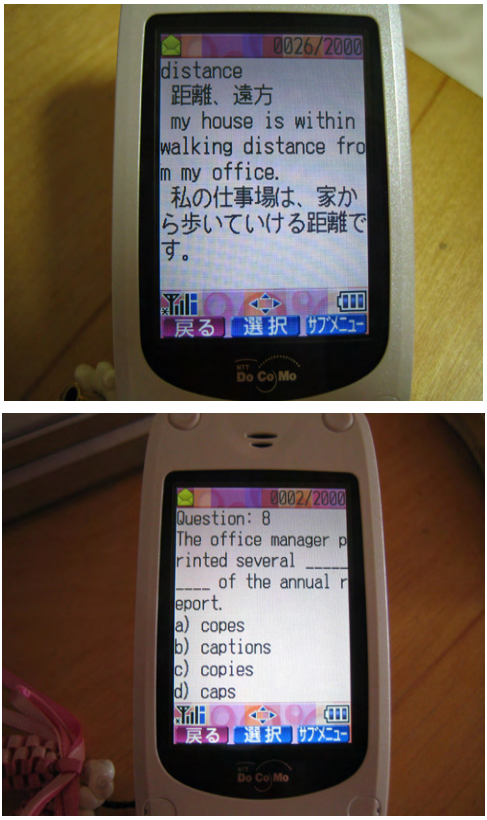

Figure 6: Mail Quiz study content and multiple choice quiz viewed on a mobile phone.

Digital white board module

Organizing or structuring ideas using concept maps is argued to be an effective means (Novak, 1990) for learners to construct their own knowledge. Because of *Moodle*'s social constructivist design, a shared digital white board complements *Moodle*'s design as well as the aims of constructivist language learning curricula. The digital white board module allows students and teachers to work together to create objects or write text on a shared digital whiteboard. Students can work in groups or individually within a course. It can be used, for example, for students to collaboratively create knowledge maps, collaborate on projects, or take group notes. This module requires Adobe *Flash Media Server*.

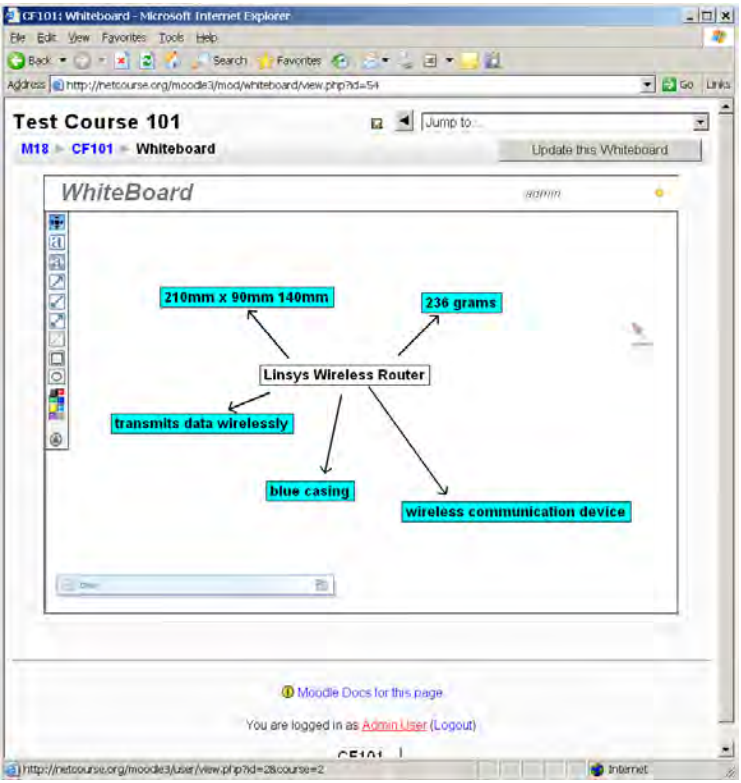

Figure 7: Adobe Flash-based shared digital whiteboard.

Student survey module

Moodle has a very effective survey module which allows instructors to collect data on student perceptions towards the quality of online coursework, but it doesn't allow individual students to create their own surveys. The tasks involved in survey projects, such as collecting, analyzing and reporting on data, work well in the social constructivist classroom. The survey module allows students to both create and administer multiple choice surveys from within a *Moodle* course.

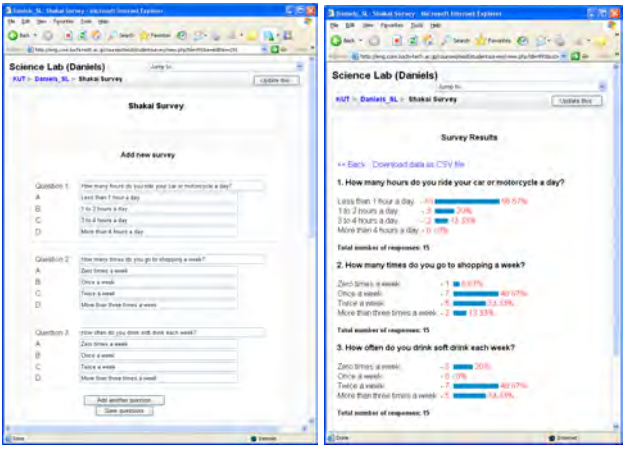

Figure 8: Student survey creation form and survey results page

Future directions

The small sample of modules described here suggests how open source CMS can be improved to better serve a specific instructional setting. The future of CMS design will continue to shift in the direction of allowing the instructor more control over the design and function of the software, rather than trying to adapt developers' designs into a curriculum. Open source systems will lead the way with extendible base cores that allow simplified methods of adding modular activities such as the ones described above.

Another important element will be the mobile platform. Typically most CMS are designed to be accessed over a standard web browser. As mobile devices become more powerful and are able to run standard web browser software such as the Apple *iPhone*'s Safari browser, learners will be able to interact seamlessly with learning content and activities both on and off campus. Several initiatives are already in progress to serve mobile learning content to mobile devices, especially in Asia, where mobile phone use outweighs personal computer use.

Content and activities can be labeled and shared as learning objects. These are reusable bits of information or activities that can be tagged and adapted to provide instruction to different learners in different environments. Wide recognition of the notion of learning objects and their integration with a wide range of CMS could lead to more individualized and productive learning experiences. While learning objects may not make any significant progress in the next decade, progress is expected in the creation of learning objects or activities that can be labeled with metadata so that they can be shared across a number of competing CMS. For example, it is now possible to share SCORM compatible content or lessons between *Moodle*, *Blackboard*, *WebCT*, *ANGEL*, and *eCollege*. As tagged learning content becomes more easily accessible and sharable across platforms, digital repositories will possess a greater degree of usefulness.

References

Adkins, S. (2005). Wake-up call: Open source LMS. *Learning circuits*. Retrieved January 28, 2008 from http://www.learningcircuits.org/2005/oct2005/adkins.htm

Astleitner, H., & Wiesner, C. (2004). An integrated model of multimedia learning and motivation. *Journal of Educational Multimedia and Hypermedia, 13*(1), 3-21.

Dougiamas, M., & Taylor, P.C. (2003). Moodle: Using learning communities to create an open source course management system. *Proceedings of the EDMEDIA 2003 Conference*, Honolulu, Hawaii.

Goodfellow, R., Lamy, M.-N., & Jones, G. (2002). Assessing learners' writing using lexical frequency. *ReCALL, 14,* 129-142.

Jonassen, D.H. (2007). A taxonomy of meaningful learning. *Educational Technology, 47*(5), 30-35.

Laufer, B., & Nation, P. (1995). Vocabulary size and use: Lexical richness in L2 written production. *Applied Linguistics, 16*(3), 307-329.

Lessig, L. (1999). *Code and other laws of cyberspace*. New York: Basic Books. Retrieved January 30, 2008 from http://www.lessig.org/content/articles/works/pcforum.pdf

Mayer, R.E. (2001). *Multimedia learning.* Cambridge: Cambridge University Press.

Novak, J.D. (1990). Concept maps and Vee diagrams: Two metacognitive tools to facilitate meaningful learning. *Instructional Science, 19,* 29-52.

Piaget, J. (1972). *The psychology of the child.* New York: Basic Books.

Prensky, M. (2001). Digital natives, digital immigrants. *On the Horizon, 9*(5), 1–2.

Vygotsky, L. (1978). *Mind in society.* Cambridge: Harvard University Press.

Key Terms & Definitions

- Generation Y refers to individuals born from 1980-1995, who are also referred to as *digital natives* because they grew up surrounded by digital technologies.
- Web 2.0 is a term used to describe the new collaborative features of the Internet.
- PHP is a server-side HTML embedded scripting language
- MySQL is a popular open source SQL database server.
- MPEG-4 is a newer compressed video standard used to deliver video over the Internet.
- 3GP is a simplified MPEG-4 video standard commonly used for video with mobile devices.
- FLV are Adobe Flash video files. Most notably, *YouTube.com* uses Flash Video.
- SCORM Sharable Content Object Reference Model is standard that packages learning content to be transferred or shared between various web-based e-learning systems.

Author Biodata

Paul Daniels is an Associate Professor of EFL at Kochi Institute of Technology in Japan. His research involves educational technology and content-based instruction. He has been actively developing activity modules for language learners and teachers that can be used with the Moodle course management system.