Development of Moodle Plug-ins that support SCORM 2004

Yosuke MORIMOTO[a*], Kiyoshi NAKABAYASHI[b], Hidenori SUGIYAMA[a] & Junji SHIBASAKI[a]
[a]*Center of ICT and Distance Education, the Open University of Japan, Japan*
[b]*Faculty of Information and Computer Science, Chiba Institute of Technology, Japan*
*morimoto@ouj.ac.jp

Abstract: We have proposed a new architecture for e-learning systems, named ELECOA, and developed Moodle plug-ins that support it. ELECOA is an architecture that fulfills both function extensibility and content interoperability, and the Moodle plug-ins we developed can run content based on ELECOA. We have also implemented the functions of SCORM on ELECOA, so the plug-ins can also run SCORM content.

Keywords: e-learning system, courseware object, SCORM, Moodle

1. ELECOA: An Extensible Architecture for e-Learning Systems

As evidenced by the open educational resources (OER) movement of recent years, there is a lot activity related to the promotion of sharing, distributing, and reusing learning content. Interoperability and reusability of learning content is key in such activities, but it is widely recognized that conventional e-learning systems and content specifications lack sufficient function extensibility. This is often because newly added functions on existing e-learning platforms may conflict with existing learning content, or it can simply be too difficult to add new functions. Even if new functions can be added without any problems, the content will not be interoperable. Sharable Content Object Reference Model (SCORM), which is the de facto standard of e-learning, has the same problem. In response to this problem, we have proposed a new architecture for e-learning systems, named ELECOA (Extensible Learning Environment with Courseware Object Architecture), which fulfills both function extensibility and content interoperability requirements (Nakabayashi & Morimoto, 2012; Nakabayashi, Morimoto, & Hada, 2010). Figure 1 shows the architecture of ELECOA. We propose using a courseware object, which is a program module that implements various educational functionalities, as a layer rather than combining it with the platform as in the conventional architecture. Content are run by the courseware objects which are assigned to them, and introducing new functions or extending existing functions is done by adding new courseware objects. Since this addition does not affect existing courseware objects, function extensibility can be assured. Interoperability can also be assured by distributing courseware objects with the content.

We have established some rules related to content structure, content packaging format, communication method between courseware objects, etc. in order to achieve the ELECOA framework. For example, ELECOA content must be structured hierarchically, which is a common structure for e-learning content. Courseware objects are assigned to respective nodes (root, branch, and leaf) of a content tree. A courseware object assigned to a node is responsible for the behavior of the sub-tree under the assigned node. This makes it possible

to implement different pedagogical strategies in different sub-trees and to distribute only specified sub-trees of content. More details are described by Nakabayashi and Morimoto (2012) and Nakabayashi, Morimoto, and Hada (2010).

Although the concept of ELECOA is independent of any programming language, we do need to choose one for use in the actual system. We chose PHP to implement the courseware objects and the platforms for running them. Courseware objects are implemented as PHP classes and are instantiated when the content is launched. In order to investigate the feasibility of ELECOA, we implemented the SCORM 2004 Sequencing and Navigation Specification (Advanced Distributed Learning, 2006) on it. Results showed that it is possible to implement a set of courseware objects that are fully compliant with SCORM 2004 SN.

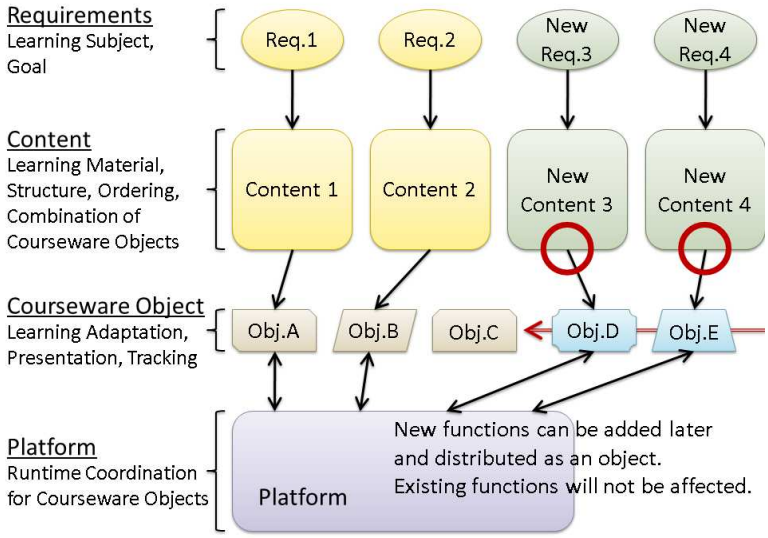

Figure 1. The architecture of ELECOA.

2. Implementation of ELECOA on Moodle

Because courseware objects are independent of platform, they can be used on any e-learning system that supports ELECOA. We have developed Moodle plug-ins that support ELECOA. As mentioned above, we also have developed courseware objects that support SCORM 2004. This means that the plug-ins can support SCORM 2004 by using the courseware objects.

There are three plug-ins: an “activity module”, a “block” (for grade reports), and a “course format”. The latter two are dependent on the activity module. The activity module, named mod_elecoa, works in the same way as mod_scorm (the built-in SCORM module of Moodle) from the point of view of the user. mod_elecoa can launch ELECOA content that are run by the courseware objects assigned to them. Users can thereby run SCORM content (both 1.2 and 2004). When the content is uploaded and added to a course as an instance of mod_elecoa, its manifest file is converted to that of ELECOA. Although mod_scorm itself is able to run SCORM content, mod_elecoa cannot; it is the function of SCORM-compliant courseware objects that are pluggable.

Figure 2 shows a screenshot while running a content on mod_elecoa. The menu area is in the upper right and the content area is in the lower right. If the content is a SCORM content (or more accurately, an ELECOA content that uses SCORM-compliant courseware objects), navigation menus are shown in the menu area and an SCO or an asset is shown in

the content area. The structure of the content and the titles and current status of the nodes are shown in the TOC.

An early version of mod_elecoa (Morimoto, Nakabayashi, Sugiyama, & Shibasaki, 2012) had performance problems that were mainly related to the speed of loading Moodle core libraries. In that early version, the Web browser communicated with the Moodle core several times per user action, e.g., clicking the "Continue" button. Each communication took several hundreds of milliseconds, which degraded the overall performance. Although we have improved the performance of mod_elecoa by reducing the number of communications with the Moodle core, further improvement is still required.

The developed plug-ins are compliant with Moodle 2.x and are distributed under the GPL via http://elecoa.ouj.ac.jp/ (Note: currently this Web site is available in Japanese only). "ELECOA Player", which is a standalone e-learning system that supports ELECOA, is also distributed under the Modified BSD License via this Web site.

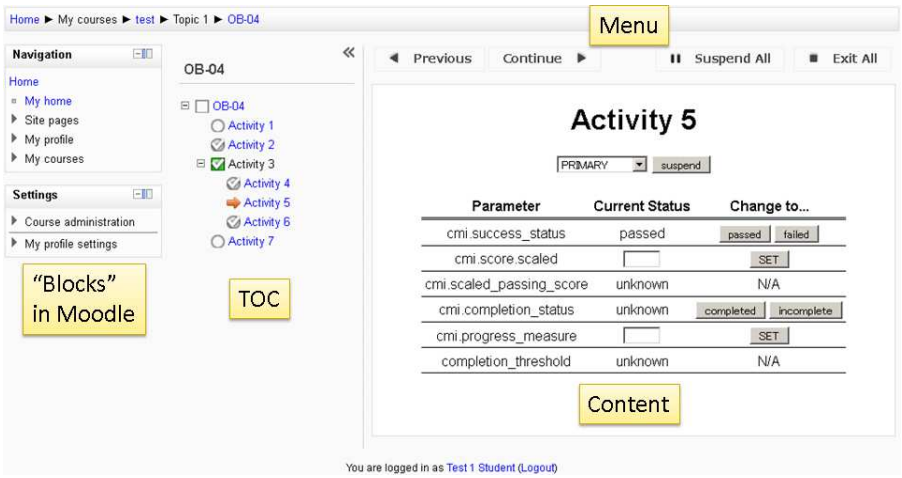

Figure 2. A screenshot of mod_elecoa.

3. Conclusion

In this paper, we described the concept of an extensible architecture for e-learning systems named ELECOA. This architecture features "courseware objects" that are program modules used to implement various educational functionalities. We were able to implement courseware objects which are fully compliant with SCORM 2004 SN. We also developed a set of Moodle plug-ins that are compliant with ELECOA. By using the courseware objects for SCORM, these plug-ins are also compliant with SCORM.

As of this writing, mod_scorm supports very few functions of SCORM 2004. The developed plug-ins should therefore be quite useful for disseminating SCORM and Moodle.

References

[1] Advanced Distributed Learning (2006). SCORM 2004 3rd Edition Specification.

[2] Nakabayashi, K., & Morimoto, Y. (2012). Design and Implementation of Object-oriented Architecture for Extensible Learner-adaptive Self-learning System (in Japanese). *Transactions of Japanese Society for Information and Systems in Education, 29(2)*, 97-109.

[3] Nakabayashi, K., Morimoto, Y., & Hada, Y. (2010). Design and Implementation of an Extensible Learner-Adaptive Environment. *Knowledge Management & E-Learning: An International Journal, 2(3)*, 246-259.

[4] Morimoto, Y., Nakabayashi, K., Sugiyama, H., & Shibasaki, J. (2012). Development of a Utilitarian UI for a Moodle Activity Module That Supports SCORM 2004 (in Japanese). JSiSE Research Report, 27(1), 11-16.