Applying an Extensible Learning Support System to Learning by Problem Posing

Kiyoshi NAKABAYASHI [a*] & Yosuke MORIMOTO[b]
[a]*Faculty of Information and Computer Science, Chiba Institute of Technology, JAPAN*
[b]*Faculty of Liberal Arts, The Open University of Japan, JAPAN*
* knaka@net.it-chiba.ac.jp

Abstract: This paper describes an investigation in which the Extensible Learning Environment with Courseware Object Architecture (ELECOA) was applied to a learning environment where problems are posed by learners. The aim of ELECOA is to provide a flexible learner-adaptive system that ensures both function extensibility and content reusability. To achieve this goal, the concept of a "courseware object" has been introduced to allow incremental implementation of various educational functionalities. Several learning environments have been based of this concept, including a self-learning environment compliant with the SCORM 2004 specification and a group-learning environment based on the Learning Design specification. This paper reports the results of a prototype implementation of ELECOA in an environment for learning by problem posing that is a mixture of self-learning and group-learning.

Keywords: Learner adaptation, extensible learning support system, courseware object, group learning, problem posing by learner

1. Introduction

Learning by question generation or problem posing is recognized as an effective way to enable learners to achieve a deep understanding of their studies (Brown and Walter, 2005). Recently, various learning environments have been designed and implemented to support problem posing by learners (Hirai, Hazeyama, and Inoue, 2009; Takagi and Teshigawara, 2006; Yu and Wu, 2012; Yamamoto, et al., 2013). These environments can be categorized into two types. In the first type, the system, equipped with certain domain knowledge on the study subject, supports and diagnoses questions generated by a learner (Yamamoto, et al., 2013). In the second type, the system provides a group learning environment in which learners can generate questions, share them with each other, and discuss how to improve them (Hirai, Hazeyama, & Inoue, 2009; Takagi and Teshigawara, 2006).

This study deals with the second type in which questions are generated in a group learning environment. As for the question generation environment, we used the Extensible Learning Environment with Courseware Object Architecture (ELECOA), which is capable of both function extensibility and content reusability (Nakabayashi, Morimoto and Hada, 2010). ELECOA is based on the concept of a "courseware object". A courseware object is a program module that is used to incrementally implement various educational functionalities. In our previous study, ELECOA was applied to several learning environments (Nakabayashi, Morimoto and Aoki, 2012), including a learner-adaptive self-learning environment compliant with the SCORM 2004 specification (ADL, 2006) and a group-learning environment based on the Learning Design (LD) specification (Koper and Tattersall, 2005; IMS, 2003). In this study, a learning environment for question generation was designed by exploiting existing courseware objects for self-learning and group-learning in combination with newly developed courseware objects for the question generation functionality.

The following section explains ELECOA and the concept of a courseware object. After that, the courseware objects for question generation and their prototype implementation are introduced. Several possible designs of learning environments for question generation and future issues are discussed in the last section.

2. Extensible Learning Environment with Courseware Object Architecture (ELECOA)

To let content designers create educational content to achieve their learning objectives without requiring deep programing knowledge, conventional self-learning systems usually use an architecture in which the content is clearly separated from the platform. The content is an aggregation of learning materials related to a certain learning subject, whereas, the platform is equipped with generic functionalities, such as learner management, learning log recording, and learner adaption, that are common to the various study subjects. The drawback of this architecture is the lack of function extensibility. If content designers or instructional designers want to create content with new educational functionalities, it is quite difficult to extend the platform once it has been implemented. This is because there is no mechanism to assure that the existing learning content for the original platform will work correctly on the extended platform.

ELECOA was proposed as a way to overcome the drawbacks of conventional systems and achieve both function extensibility and system interoperability (Nakabayashi, Morimoto and Hada, 2010). ELECOA is characterized by its modular system architecture shown in Figure 1. It employs the concept of a “courseware object”, a building block that implements various educational functionalities, such as sequencing strategies of learning materials. As shown in Figure 1, the courseware object is clearly separated from the platform, thereby enabling incremental function extension by adding new courseware objects. Since the existing courseware objects and new courseware objects do not interfere with each other, the existing content will work correctly after the function extension.

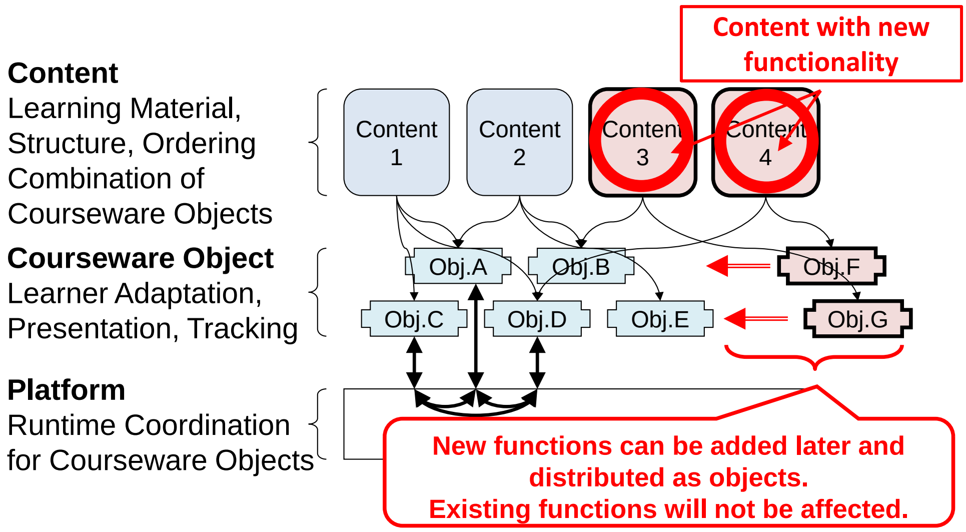


Figure 1. ELECOA framework.

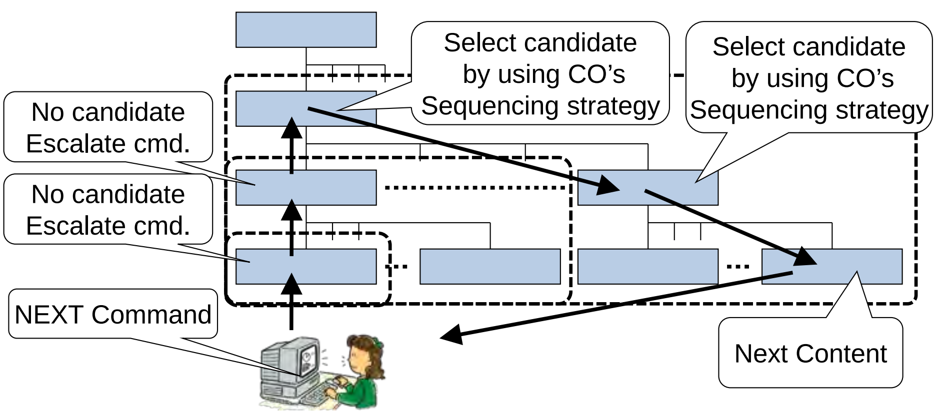


Figure 2. ELECOA for content with a tree structure.

ELECOA has been applied to both self-learning and group-learning environments with tree-like learning-flow control structures (Nakabayashi, Morimoto and Aoki, 2012; Nakabayashi and Morimoto, 2013). In these applications, courseware objects are allocated to the tree nodes, as shown in Figure 2. Each courseware object then manages the sequencing of the sub-tree under its node. According to the sequencing strategy implemented in it, the courseware object is responsible for selecting the most suitable node from among its child nodes. This sort of configuration makes it possible to implement different pedagogical strategies in different sub-trees. To achieve interoperability between courseware objects created by various implementers, the communication between courseware objects is limited to only between parents and children. The basic communication patterns and interface between the courseware objects have been designed within this limitation, and defining these basic communication patterns and the interface maximizes the interoperability of courseware objects. As the example of a self-learning environment, the SCORM 2004 specification, a standard for learner-adaptive content, has been implemented according this configuration (Nakabayashi, Morimoto and Hada, 2010). The implementation was successfully examined against test cases of the SCORM 2004 3rd edition test suite.

To apply ELECOA to an LD-specification-based group learning environment, we exploited the fact that the LD specification, which is capable of formally describing a group learning sequence, uses a tree format, like the self-learning environment (Nakabayashi, Morimoto and Aoki, 2012; Nakabayashi and Morimoto, 2013). The implementation is shown in Figure 3. A tree-like learning flow control is assigned to each individual learner. The courseware object allocated at each tree node controls the learning sequence of each individual learner by taking into account the learner's status such as learner's level and accomplishments. This is the same manner as the original ELECOA for self-learning. Thus, it is possible to implement a learning path control based on each individual's status as required by the LD specification. In addition, a courseware object can reflect the statuses of multiple learners in order to select each individual learning path. This is possible by designing the courseware object to be able to exchange information with other courseware objects assigned to other learners. This means that each learner's learning path can be controlled by his or her own learning status as well as those of other learners. Regarding communication tools, these are associated with the leaf node of the hierarchical structure as learning resources. This is a requirement of the LD specification, and at the same time, it is consistent with the original ELECOA self-learning environment in which learning resources are assigned to the leaf nodes of the content tree.

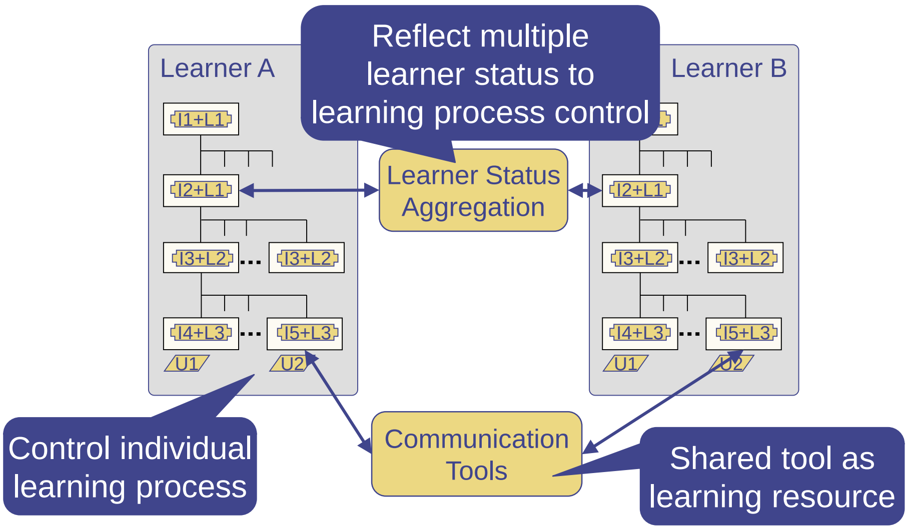


Figure 3. Implementation of group learning environment with ELECOA.

3. Application of ELECOA to Problem Posing

This section discusses an application of ELECOA to a problem-posing environment. The target is an environment in which a group of learners generates questions, then other learners and teachers share, review, and improve the questions. The generated questions are sometimes utilized as course materials (Hirai, Hazeyama, and Inoue, 2009; Takagi and Teshigawara, 2006). It is not in the scope of this study but rather a potential future work item to implement the function to support or diagnose the problem-posing activities by using domain knowledge about the specific subject. Rather, this study proposes a generic group learning framework for problem-posing environments based on ELECOA courseware objects that were designed in accordance with common basic function requirements extracted from existing problem-posing environments which were implemented ad-hoc.

Problem-posing group learning environments require three basic functions:

(1) Control of discussions associated with problems,
(2) Generation and modification of questions as well as generation of associated discussions, and
(3) Utilization of questions as materials

The first requirement is common to the group learning environment described in the previous section. In the problem-posing environment, however, the role of the learner who creates the problems may be different from other learners and teachers. There can be a pre-defined sequence of discussions that is used to control the discussions associated with each generated problem. The second requirement is specific to problem-posing group learning environments. The original ELECOA architecture assumes that the learning control tree and associated learning resources are defined before the start of learning. Problem-posing group learning environments require a capability that lets learners generate questions dynamically while they are learning and to share and modify these questions based on discussion. The third requirement is that the generated questions should be able to be dynamically utilized as conventional self-learning materials.

Figure 4 depicts the ELECOA-based problem-posing group learning environments designed according to the above requirements. The “Problem list” courseware object has “problem” courseware objects as its child nodes. Each problem courseware object has a “problem content” learning resource and “discussion” courseware object as its children. The “Discussion” courseware object has “discussion content” learning resource. This tree structure is assigned to every learner participating in the group learning.

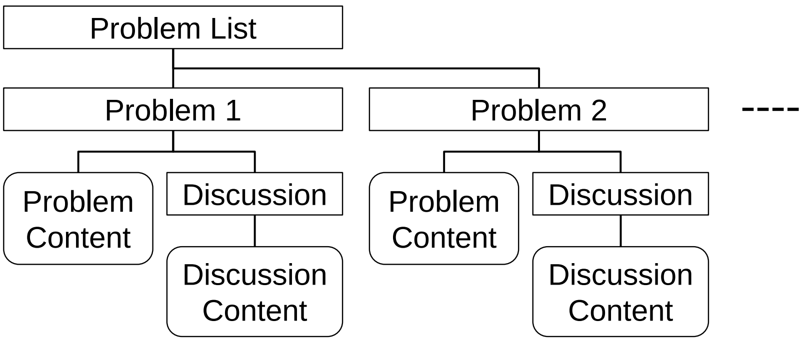


Figure 4. Implementation of problem posing group learning environments with ELECOA.

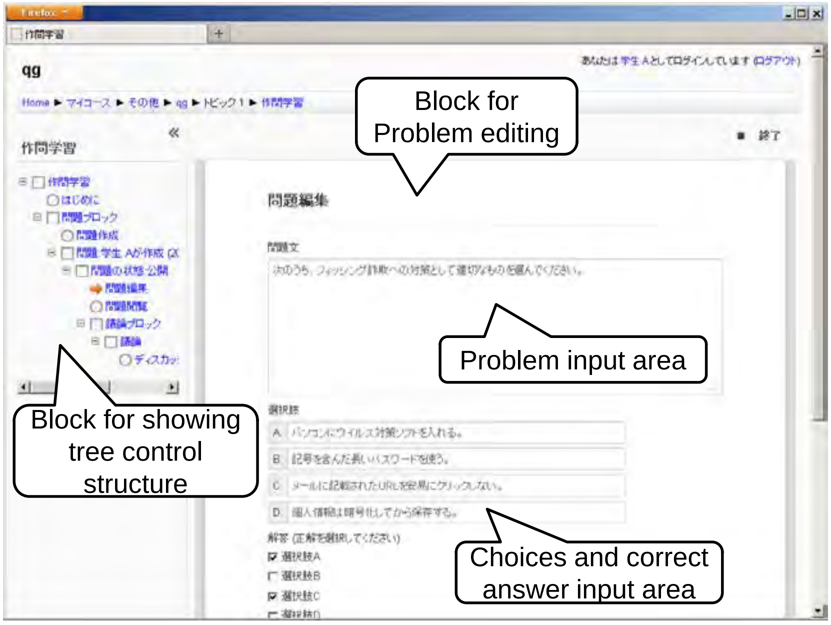
The problem list courseware object accepts and executes commands from learners, including “create new problem”, “list problems” and “select problem”. In particular, the “create new problem” command generates a new problem courseware object as the child of the problem list courseware object. The problem courseware object executes commands such as “edit problem content”, “view problem content” and “submit problem”. The discussion courseware object executes commands such as “submit comment” and “view comment”.

The problem courseware object and discussion courseware object manage the state transition table to control the problem-posing learning scenario. For example, a certain problem-posing learning strategy may require the following learning process:

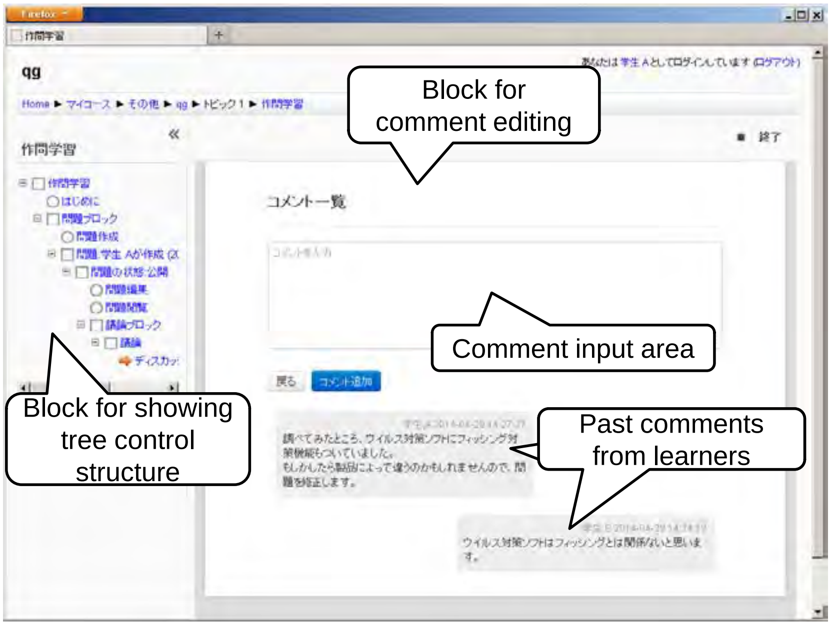
(1) A learner creates a new problem,
(2) Other learners answer the problem,
(3) Other learners evaluate and give comments on the problem,
(4) The learner who created the problem modifies the problem,
(5) The learner may ask for comments from the teacher before he/she modifies the problem,
(6) These steps are repeated until there are no comments to be reflected.

In this learning scenario, the learner who created the problem must modify question by him-/herself. There can be other learning scenarios in which every learner has an opportunity to give comments and modify the problem at any time. It is possible to implement an appropriate state transition table to be managed by each courseware object according the chosen learning scenario.

Figure 5 shows screenshots of the prototype implementation. The left area of each window is the tree control structure corresponding to the one shown in Figure 4. From this area, the learners can take commands to create new problems, modify problems, and participate in discussions. The prototype was developed as a Moodle plugin so that the Moodle learner group assignment is inherited by the developed environment.



a) Problem editing window



b) Comment editing window

Figure 5. Screenshots of prototype implementation.

4. Conclusion

This paper described an investigation and prototype application of ELECOA to a learning environment for problem posing by learners. The aim is to provide a generic system framework for problem-posing environments. Several courseware objects were developed according to the requirements and a prototype system was implemented. In the future, we will deploy this system in an actual classroom environment and assess how it works. It will also be necessary to improve the communication protocol between newly developed courseware objects so that they are consistent with the existing self-learning courseware objects to develop an unified learning environment capable of integrating self-learning and group learning.

Acknowledgements

This work was supported by Grants-in-Aid for Scientific Research by the Japan Society for the Promotion of Science (Kakenhi 26280128 and 26242013).

References

Advanced Distributed Learning (2006). *Shareable Content Object Reference Model SCORM® 2004* (3rd ed.)

Brown, S.I., & Walter, M.I. (2005). *The art of problem posing* (3rd ed.). New Jersey: Lawrence Erlbaum Associates.

Hirai, Y., Hazeyama, A., & Inoue, T. (2009). Assessment of Learning in Concerto III: A Collaborative Learning Support System Based on Question-posing, In *Proceedings of CATE2009* (pp. 36-43). Calgary, Canada: ACTA Press.

IMS Global Learning Consortium (2003), *IMS Learning Design Version 1.0 Final Specification*.

Koper, R. & Tattersall, C. (2005), *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*, Springer.

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

Nakabayashi, K., Morimoto, Y., & Aoki, K. (2012). Application of Extensible Learning Support System Architecture to Collaborative Learning Environments, In *Proceedings of the 12th IEEE Intentional Conference on Advanced Learning Technology* (pp. 69-73), Rome, Italy: IEEE Computer Society.

Nakabayashi, K. & Morimoto, Y. (2013). Investigation on Function Extension of Extensible Learning Support System Architecture to Group Learning Environment, In *Proceedings of the IEEE International Conference on Teaching, Assessment and Learning for Engineering* (pp. 335-339), Bali, Indonesia: IEEE Computer Society.

Takagi, M., & Teshigawara, Y. (2006). A WBT System Enabling to Create New or Similar Quizzes Collaboratively by Students. In *Proceedings of ICET2006* (pp. 263-268). Calgary, Canada: ACTA Press.

Yamamoto, S., Hashimoto, T., Kanbe, T., Yoshida, Y., Maeda, K., & Hirashima, T. (2013) Interactive Environment for Learning by Problem-Posing of Arithmetic Word Problems Solved by One-step Multiplication. In *Proceedings of the 21st International Conference on Computers in Education ICCE 2013* (pp. 51-60). Bali, Indonesia: Uhamka Press.

Yu F-.Y. & Wu C-.P. (2012) Student Question-Generation: The Learning Processes Involved and Their Relationships with Students’ Perceived Value. *J. Research in Education Science*, *57*(4), 135-162.