Effective Courseware Design and Analysis Using SMARTIES

Jaemu Lee*, Yusuke Hayashi** and Riichiro Mizoguchi***

(Received 18 June 2012 and accepted in revised form 25 February 2013)

Abstract Much courseware has been developed recently; but not all of it is of high quality, nor is all of it based on educational theories. Our purpose in this study is to support the design of high quality courseware that reflects instructional strategies and supports the design easily, especially for novice designers. The instructional model ontology of expert knowledge and experience is constructed using SMARTIES, an authoring tool. Designers can develop their courseware efficiently using this instructional model ontology. We demonstrate the efficacy of the courseware development through an experimental study that included experimental and comparative groups. We found from survey results of the experimental group that designing courseware through the instructional model ontology of SMARTIES improves the quality of the courseware and it is much more helpful for inexperienced designers. Overall, our method led to the design of courseware that was more concrete and more faithful to instructional strategies.

Keywords: courseware design, instructional model, ontology for education, direct instructional model, instructional strategy

1. Introduction

As information and communication technologies are improved, the area of e-education is becoming more active; and, along with this activity, the applications of courseware have become more widespread. Courseware has several advantages: it provides individualized learning and improved learning, and has an effective time-cost performance. It is, however, also true that developing high quality courseware is difficult, and it takes a long time to develop great courseware; therefore, proper tools have been developed to help design courseware more easily. Most of these tools, however, have failed to make any significant contribution to the development of courseware. The reason is that the existing tools support education only just roughly, and can support neither the instructional models nor the instructional strategies that have been created to attain educational goals. For the successful design of courseware, and in order to accomplish learning goals, we need to deeply understand the educational content and the effective instructional strategies that reflect instruction and learning principles. With an instructional model included in the courseware, therefore, the courseware will exhibit better effectiveness. In addition, with courseware authoring tools supporting an instructional model, designers can develop better courseware. Although there are some knowledge representation tools based on mind tools that help designers develop instructional models, most of them do not support instructional models as well.

We have developed an authoring tool called SMARTIES for the representation of education knowledge. SMARTIES is a tool powerful enough to show the overall structure of knowledge based on ontology. It also has a function for sequencing e-learning content. Not many suggestions, however, have been made for its actual application so far. With implementation of an instructional model the existing functions for sequencing content would be more useful and be supplemented with instructional strategies.

Our purpose in this study is to support the design of high quality courseware that reflects instructional strategies and supports designing easily and efficiently, especially for novice designers. We construct the instructional model ontology through our ontology-based, educational authoring tool, SMARTIES. We demonstrate the efficacy of the courseware development through an experimental study using two groups. Students majoring in computer education at a teachers’ college in Korea participated in the study; they were divided into an experimental group and a comparative group. We analyzed the opinions of the students in the experimental group through a closed-ended survey and interviews.
2. Related Studies

We reviewed advanced research on courseware development tools, and some development tools, such as courseware generation, courseware sequencing, and courseware design support tools. We found that work by Ullrich and Melis\(^9\) was representative of research on courseware generation. Unlike other courseware generation approaches, however, their research generated structured courses that can be adapted to a variety of learning goals and to the learners’ competencies. Their research also described basic methods for course generation, which were used to formalize seven different types of courses. Finally, their research had the merit of supporting adaptive courseware from the viewpoint of instructional strategies, but it was limited because it did not support instructional design strategies using instructional models.

Research on courseware sequences, such as conducted by researchers like Karampiperis\(^{10}\), is to a large extent about sequencing, such as course organization in intelligent tutoring systems, or learning objects for the standard content. In particular, however, there has been a failure to represent instructional strategies and instructional models\(^{10,11}\).

There is, however, some other research that helps designers develop courseware through knowledge tools. Nkambou et al.\(^{12}\) have developed their Curriculum Representation and Acquisition Model (CREAM) which supports the creation and organization of the curriculum through a domain pedagogical approach and the didactic aspects of the instruction. Ritter and Blessing\(^{13}\) have presented their Visual Translator, which is a tool designed to support the development of educational systems. This approach that helps designers develop courseware through knowledge tools leads to the creation of a Concept Map-like representation of the course description, with some concepts corresponding to the topics in the courseware, and others providing elaboration of the relationships among the topics. Coffey\(^4\) has developed a software tool that supports this method of course development, and he described an example of the utilization of this tool to convert a Concept Map into a course description and he elaborated on how the tool fosters course components. These studies help design courseware with meta tools, but merely expanding the concept of mapping is not enough to support the entire trend of learning with instructional strategies. In addition, information expressed in one node is limited and its visual function is not satisfactory.

From our review, we discovered a need for a new method to support instructional models and help design them easily.

Using an instructional model that applied SMARTIES, we attempted to help designers become familiar with thinking and developing creative ideas, while showing them the entire structure and learning order of the courseware.

3. Courseware Design through the Instructional Model Ontology

3.1 SMARTIES overview

This study builds an instructional model ontology using SMARTIES. The SMARTIES system is a scenario-based, authoring tool devised by us\(^{7,8}\). It uses the OMNIBUS ontology, which contains stored educational theories, in order to build a decomposition tree. The decomposition tree consists of the concept Instruction Learning event (I-L event) and why-knowledge defined in the OMNIBUS ontology. In this study, we represent a decomposition tree as a sequence and combination of I-L events and why-knowledge for each instructional model using SMARTIES as shown in Figure 1. The items in the ellipses such as ‘Introduction’, ‘Review’, ‘Objective’, and ‘Motivation’ are I-L events. Each I-L event can take one or more sub I-L events and the events are in sequence. The ‘Introduction’ is achieved by the and relationship of ‘Review’, ‘Objective’, and ‘Motivation’, and this relation is called ‘why-knowledge’. The why-knowledge is represented by and/or relationships. We newly constructed the instructional model ontology that is the process-oriented model for a one-unit lesson, and we suggest a new method to assist in the instructional design of courseware.

3.2 Instructional model ontology construction

We constructed the instruction ontology to utilize instructional models. The knowledge of instruction ontology was constructed with SMARTIES, made to support the design of courseware. In addition, when this knowledge was used to construct the ontology, the instructional models that developed educational theories
were applied to the large concepts of the instructional models. The educational theories, however, describe generally large concepts. They do not clearly describe the subordinate concepts and detailed processes. We obtained the subordinate concepts by analyzing class guideline instructional strategies of existing lesson plans, and we also interviewed expert teachers for each model. Our long-term experience in computer teaching methods was valuable for determining the subordinate concepts for the construction of the ontology.

In this section, the ontology for a direct instructional model is explained through examples.

Figure 2 is a decomposition tree showing the overall structure of a direct instructional model made using SMARTIES. It shows the overall learning order of the direct instructional model. The learning goals are divided into such steps as ‘Introduction’, ‘Development’, and ‘Consolidation’ applying educational theories. This knowledge from these theories does not supply enough information for courseware design. We therefore saved more detailed knowledge into the instructional model ontology by extracting knowledge from our experiences and those of other experts. The ‘Introduction’ is divided into three steps: ‘Review’, ‘Objective’, and ‘Motivation’. The ‘Development’ is divided into two steps: ‘Explanation’ and ‘Practices’. Last, through ‘Consolidation’, learners can learn to ‘Test’ and make a ‘Summary’. As is seen in Figure 2, we constructed knowledge about instructional models through SMARTIES. Designers can establish their own decomposition trees by selecting, modifying, or adding content to the instruction ontology constructed; thus, a general framework for an instructional model is established, indicating how to develop learning content in order.

In Figure 3, the overall structure is decomposed to describe Figure 2 in more detail, showing the ‘Development’ step. The ‘Explanation’ and ‘Practices’, parts of the ‘Development’ step, are subdivided into smaller steps. The ‘Explanation’ step is not subdivided, but the ‘Practices’ step is subdivided into smaller steps: ‘Applying’ and ‘Self-Reflection’ in the present study. In addition, the ‘Applying’ step is divided once again into subordinate steps: ‘Guiding’ and ‘Performing’. In Figure 2, ‘Development’ is divided into ‘Explanation’ and ‘Practices’; but in Figure 3, it is additionally subdivided into smaller steps: ‘Explanation’, ‘Guiding’, ‘Perform’, and ‘Self-Reflect’ in the suitable order for learning.

When courseware is designed, the order of the courseware content is developed through steps divided to the lowest level. Specifically, building a decomposition tree in each step concretely indicates the described steps in order to accomplish smaller-unit learning goals. Accordingly, what is seen in a decomposition tree is, in fact, information about the framework of the learning-development processes and the instructional strategies. As shown in both Figures 2 and 3, low-level steps decomposed by means of a decomposition tree are expressed while being mapped into each screen of the storyboard.
3.3 Courseware design process

This section describes a method of designing through SMARTIES. Most present-day courseware designers develop courseware by using flowcharts. Instead, we used SMARTIES; the overall framework is expressed in a decomposition tree through SMARTIES, as shown in Figures 2 and 3 and the courseware was designed to follow the order appearing in the detailed decomposition tree.

If the decomposition tree in Figure 3 is to be that of
an instructional model for learners to accomplish their own learning goals, courseware needs to be designed whose learning content can be developed in subdivided steps: ‘Explanation’, ‘Guiding’, ‘Perform’, and ‘Self-Reflect’ steps in that order. Finally, as shown in each scene appearing on the storyboard of the courseware, instructional strategies appearing in the decomposition tree are realized through the screens of the courseware; that is, each node of the decomposition tree is mapped into the screens of the courseware.

Figure 4 shows one of the scenes on the screen. All the information needed for the screen of the courseware is expressed on the storyboard: courseware topics, courseware-applied instructional models, steps in the instructional model, instruction procedures applied, courseware screen design, and courseware instructional strategies. In particular, the learning content is developed essentially through the detailed steps appearing in the decomposition tree.

4. Application and Evaluation

4.1 Comparison of courseware

4.1.1 Analysis subjects and analysis methods

The research participants were divided into an experimental group and a comparative group and we comparatively analyzed the courseware based on the direct instructional models designed by the research participants. The experimental group was asked to design their courseware with SMARTIES, while the comparative group designed their courseware using traditional methods. There were 31 participants in the experimental group and they took this class in the second semester of 2011; the comparative group had 33 participants, and they took this class in the first semester of 2011. Furthermore, this experimental study attempted to provide the same learning environment and conditions for the experimental and comparative groups.

We compared the courseware developed by participants of each group. Both groups designed courseware based on direct instructional models, and we found some differences, depending on whether the participants had used SMARTIES in designing their courseware. Then we comparatively analyzed the courseware content with some questions; for example, ‘Does the courseware have rich content?’ ‘Is it concrete, systematic, and logical?’ ‘Is the content well-balanced?’ and ‘Does it have proper multimedia expression?’.

4.1.2 Comparison and Analysis of Contents

Some of the differences that we identified are summarized in Table 1. We selected one example among the courseware projects that were developed in each group and explain it below.

(1) Example from the Comparative Group

When the participants in the comparative group designed their courseware, they had to provide an outline table for the initial page as shown in Figure 5 so that others might understand the overall structure of their courseware more easily.

From Figure 5, it was found that the comparative group developed courseware by faithfully applying a direct instructional model. In particular, the group designed the courseware while applying the upper levels of a direct instructional model, ‘Introduction’, ‘Explanation’, ‘Demonstration’, ‘Question’, ‘Practices’, and ‘Consolidation’. Most existing courseware is developed by following the steps of a general model: ‘Introduction’, ‘Development’ and ‘Consolidation’, without applying an instructional model; however, the courseware developed by the comparative group was not especially concrete since the participants in the group applied only the upper levels of the direct instructional model.

From Figure 5, it was found that the comparative group developed courseware by faithfully applying a direct instructional model. In particular, the group designed the courseware while applying the upper levels of a direct instructional model, ‘Introduction’, ‘Explanation’, ‘Demonstration’, ‘Question’, ‘Practices’, and ‘Consolidation’. Most existing courseware is developed by following the steps of a general model: ‘Introduction’, ‘Development’ and ‘Consolidation’, without applying an instructional model; however, the courseware developed by the comparative group was not especially concrete since the participants in the group applied only the upper levels of the direct instructional model.

Figure 6 shows how to design a storyboard with the content of the ‘Explanation’ step. Although courseware designers in the comparative group tried to express their storyboard as much as possible, while being faithful to each step of the direct instructional model, they failed to concretely describe the content of each step since they
focused on describing only the upper levels of the direct instructional model.

(2) Example from the Experimental Group

Before designing their storyboard, the students in the experimental group had to provide a proper instructional model frame fit for their learning goals, as in the decomposition tree shown in Figure 7. At that time, they could either create a decomposition tree by using the instruction ontology constructed by the authors or express the instructional strategies they intended as courseware designers. Designers in the experimental group modified some of the instructional strategies to their own taste by using design instructional strategies from the direct instructional model ontology constructed by us. Figure 7 shows the decomposition tree only in the ‘Development’ step. The designer in the experimental group subdivided the ‘Explanation’ step into smaller steps: ‘Overview’ and ‘Example’. The next step was the ‘Demonstration’ step; and ‘Reflect’ was subdivided into ‘Performing’ and ‘Feedback’. The ‘Question’ step was also subdivided into ‘Self-Learning’ and ‘Self-Reflect’. The designer in the experimental group mapped out each step on the storyboard, based on the established decomposition tree.

Figure 8 shows an outline table summarizing the storyboard developed by the designer in the experimental group. This outline matched the decomposition tree of Figure 7 for the experimental group. Compared to the outline table from the comparative group, shown in Figure 5, that of the experimental group in Figure 8 was found to be subdivided in more detail, and the instructional strategies were more precisely expressed in each step; for instance, while the outline table made by the comparative group (Figure 5) consists of the ‘Introduction’ step only, that made by the experimental group is more precisely composed of subordinate steps: ‘Recall prior learning’, ‘Connect prior learning’, ‘Learning objective’, ‘Learning necessity’, ‘Learning sequence’, and ‘Motivation’, in that order. In both the ‘Development’ and ‘Consolidation’ steps, the example courseware from the experimental group was found to be more concretely designed with smaller units. It was

Table 1. Comparison between the Comparative and Experimental Groups.

| Factor                | Comparative group                                                                 | Experimental group                                                                 |
|-----------------------|----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| Instructional strategy? | They tried to reflect courseware development principles, but did not represent the instructional strategy fully. | Most of the courseware represents the instructional model’s process.               |
| Concrete?             | Representation of content is by abstract concepts, and each step is not represented concretely or in detail. | Representation of content is concrete and in detail through the decomposed instructional model. |
| Rich content?         | 30% of the courseware was developed with rich content.                            | 80% of the courseware was developed with rich content.                              |
| Balanced?             | There was a 50% relationship between well- and wrongly-balanced organizations.    | The flow of the courseware was natural and balanced.                                |
possible to design courseware with such smaller units because the designer developed the content through mapping after establishing a framework from the decomposition tree as shown in Figure 7.

Figure 9 shows courseware in the ‘Overview’ step of Figure 7. Figure 10 shows the ‘Explanation’ step with examples. The courseware from the comparative group had no explanation for each unit as shown in Figure 6, but that of the experimental group was explained while being divided into ‘Overview explanation’ and ‘Explanation with examples’; since it was explained while being divided into smaller units, the purpose of each step could be expressed more clearly.

The following is a comparison of the experimental group (proposed method in this study) and the comparative group (traditional method).

The experimental group built frameworks and then designed their content while the comparison group designed their content directly. The experimental group was, therefore, given direction for the design, and they created the content easily and fluidly.

Second, while the comparative group’s design depended on their ideas, the experimental group’s design used the instructional model ontology; as a result,
novice designers could share experts’ knowledge from the ontology.

Third, while the comparative group did not represent the instructional strategy well, the experimental group was able to apply the decomposed instructional model, and was able to represent more strongly the instructional strategy in their courseware.

Fourth, while the comparative group designed without considering the decomposed goal, the experimental group expressed the content through the small decomposed goals of the instructional model; thus, this content was built concretely and in detail.

Lastly, since the courseware from the experimental group subdivides the purpose of each step and expresses subordinate steps on the storyboard, its content was developed more systematically and was more enriched than that of the comparative group.

4.2 Closed-ended survey

To discover what the participants thought about the courseware that they developed through SMARTIES during the experimental research, we conducted a closed-ended survey of the experimental group 29 participants gave replies and the responses are summarized in Table 2; the survey data were collected in class when the semester was almost complete.

Q-1 Do you think that it is important to comply with the procedures of an instructional model when designing courseware?

Six (20%) of the 29 respondents expressed the opinion that it is very important to do so. This finding indicates that most of the students in the experimental group seemed to find it important to introduce instructional models for the development of courseware.

Q-2 Do you think that making a decomposition tree is helpful in designing courseware?

The average value of the survey responses was 4.0, which indicates that most of them found it helpful to make a decomposition tree before designing the courseware.

Q-3 Had you ever designed courseware by complying with the detailed procedures of an instructional model before learning about a decomposition tree?

The average value of the survey responses was 2.2, which indicates that most of them had never designed courseware by complying with the procedures of an instructional model. In addition, the finding indicates that

| Classification                                      | Ave. | SD | Not at all | No | Normal | Yes | Well |
|-----------------------------------------------------|------|----|------------|----|--------|-----|------|
| Is the instructional model important for the courseware design? | 4.0  | 0.6| 0          | 0  | 5      | 18  | 6    |
| Is the decomposition tree useful?                   | 4.0  | 0.5| 0          | 0  | 4      | 21  | 4    |
| Have you applied a decomposition tree before?       | 2.2  | 0.5| 1          | 20 | 8      | 0   | 0    |
| Is there a match between the decomposition tree and the courseware? | 4.2  | 0.8| 0          | 1  | 3      | 14  | 11   |

Figure 9. Example of the ‘Overview’ Step.

Figure 10. Example of the ‘Explanation’ Step.

Table 2. Opinions about Decomposition Tree Use for Courseware Design.
most of them came to design courseware by complying with the procedures of an instructional model through this class only and for this semester.

Q-4 Is each step of the decomposition tree matched with each screen of the courseware?

The average value of the survey responses was 4.2, which indicates that most respondents designed courseware while complying with the instruction model ontology.

From the responses to the closed-ended questions, we found that most of the students in the experimental group thought it important and useful to make a decomposition tree and design courseware based on it. In addition, most of them never used such a method in preparing their own courseware; and, through this experiment in developing courseware, they have started to design courseware while applying SMARTIES faithfully.

4.3 Interview results

To reflect the students’ opinions when evaluating their final assignments, we conducted a personal interview with each participant. During the interviews, the students in the experimental group expressed their opinions about their designs while examining the decomposition trees and courseware that they had made. They also expressed opinions about the process of using SMARTIES. The differences were clear between those who designed great decomposition trees and those who did not. The experimental group consisted of students majoring in computer education, and they generally expressed less difficulty in using SMARTIES. The results of the interviews are summarized as below.

Most of the students in the experimental group said that using SMARTIES made their courseware more detailed and concrete, compared to courseware developed using traditional methods. They said that they found it difficult to express content when designing courseware, but when making a decomposition tree and expressing courseware by complying with the related procedures, they discovered that using SMARTIES was helpful when describing expressions in their courseware and developing the content. Consequently, the decomposition tree helped them complete the courseware more systematically and naturally. In particular, since they were able to see the overall shape, this new system was very helpful in designing better courseware. In addition, initially, as they made their framework for the design of the courseware, they found that there was less content to be modified; but, when necessary, it was easier to modify.

About 20% of the students in the experimental group voiced negative opinions. Some said that it was not worthwhile using SMARTIES since they found it difficult to use, while others pointed out the problem that it was difficult to develop flexible content since the tool provides standardized courseware only.

5. Discussion

As a result of having students design courseware through SMARTIES and then analyzing the opinions of the students about the task, we found that the following need to be discussed.

First, using SMARTIES, courseware can be systematically designed with its subordinate content enriched. At the same time, courseware designers may lack a certain amount of flexibility when depending too much on the typical framework. Both opinions are valid. It is necessary, therefore, for designers to modify the decomposition tree expressions in consideration of their learning goals. In addition, it is also necessary for courseware designers to focus on learning goals even when developing their decomposition tree.

Second, courseware designers may depend too much on instructional models. Some students expressed the opinion that it was difficult to design courseware that did not match the provided instructional model. In general, an instructional model is built into a decomposition tree through SMARTIES, and then the courseware is designed to fit the decomposition tree. The effectiveness of a decomposition tree is much higher, however, when an instructional model is not applied. This is because instructional strategies can be expressed in the decomposition tree at each step; and thus be much more effective in the use of SMARTIES.

Third, in the process of using SMARTIES, (unlike using other existing methods) it may initially take a longer time to make the decomposition tree and design the courseware based on it, but since designing the courseware is based on the framework provided by the decomposition tree, courseware designers can spend much less time on arranging their ideas. That is, it initially takes a longer time to make a decomposition tree, but less time is spent on designing courseware after-
wards; thus, overall there is no great time difference. If designers design more courseware using the same model, they would have to modify the existing decomposition tree just a little, and this would help them spend even less time. Furthermore, when sharing decomposition trees already made by developers, designers can develop courseware more efficiently.

Fourth, the instructional ontology that was built into SMARTIES can be reused by experts and designers. Therefore design can be done effectively and time can be saved. The system is especially helpful for novice designers.

What has been found while teaching SMARTIES to pre-service teachers is that it is necessary to spend some time in teaching them some of the functions of SMARTIES, and it is necessary for novice designers to provide a decomposition tree in connection with their own learning goals. The students in the experimental group were thought to have had sufficient knowledge about designing courseware, and they were taught how to practically design courseware with SMARTIES over only six weeks; it was discovered, however, that it should have been about a ten-week course. In addition, in the middle of the course, it was found that feedback was very important to the students in the experimental group. Due to time constraints, however, we were unable to provide all the help that was desired.

6. Conclusion

In this study, we discovered that designing courseware through the already constructed instruction ontology is more effective than designing courseware through traditional methods.

Participants constructed eight kinds of instructional ontology models by using the SMARTIES authoring tool. For the overall knowledge of the instructional models, the ontology was constructed on the basis of educational theories. Moreover, we incorporated our teaching experience and knowledge of instructional strategies into the ontology to make the subordinate instructional strategies more concrete. By using this instruction ontology, the students in the experimental group were instructed to design courseware and were able to develop courseware more easily by sharing the expert knowledge of the ontology. Furthermore, this authoring tool will be very helpful for novice designers.

One of the most positive effects of using SMARTIES is that it enabled students in the experimental group to develop courseware more systematically and in a more balanced manner while making the overall structure more visual. Furthermore, it also helped them to develop courseware that was more precise and more organized. In particular, most of the students who designed their courseware in traditional way tended to neglect instructional strategies while focusing on multimedia expressions and conveying learning content; but, those using SMARTIES supplemented their lack of instructional strategies somewhat by reflecting instructional models in the courseware effectively. Expressing a matching method from the physical layer of SMARTIES makes it possible to develop ideas while actually screen designing the courseware. In addition, the visual function of SMARTIES helps design courseware which is well-balanced, and whose overall order is well-organized.

With respect to future research, it will be necessary to develop tools for the design of efficient courseware and tools which can reduce the gap between the decomposition tree and the courseware content. Moreover, with more examples of tools like SMARTIES, better instructional ontology can be designed, further supporting user designs. Finally, we strongly feel that it is appropriate to expand this design method into actual development outcomes.

References

(1) Stoll, P. J.: “Develop Effective Environmental Training”, Chemical Engineering Progress, Vol. 97, pp. 42–48 (2001).
(2) Fuller, D., Ochoa, S. and Baloian, N.: “A Model for Courseware Development (MCD)”, in Proc. of the Society for Information Technology & Teacher Education International Conference 2001, ed. Price, J., pp. 2320–2323, AACE, VA (2001).
(3) Ochoa, S., Pino, J. A., Baloian, N. and Fuller, D.: “ICE SEE: A Tool for Developing Engineering Courseware”, Computer Applications in Engineering Education, Vol. 11, No. 2, pp. 53–66 (2003).
(4) Coffey, J. W.: “A Meta-cognitive Tool for Courseware Development, Maintenance, and Reuse”, Computers & Education, Vol. 48, No. 4, pp. 548–566 (2007).
(5) Shivkumar, S.: “Strategies for Improving E-Learning Effectiveness”, International Workshop on E-Learning for Adult Continuing Education and Extension, CUSAT, India, pp. 1–8 (2006).
(6) Uden, L.: “Courseware Engineering Methodology”, J. of Computing in Higher Education, Vol. 14, No. 1, pp. 50–66 (2002).

(7) Hayashi, Y., Bourdeau, J. and Mizoguchi, R.: “Using Ontological Engineering to Organize Learning/Instructional Theories and Build a Theory-aware Authoring System”, International J. of Artificial Intelligence in Education, Vol. 19, No. 2, pp. 211–252 (2009).

(8) Hayashi, Y., Bourdeau, J. and Mizoguchi, R.: “SMAR TIES: A Theory-aware and Standard-compliant Authoring System”, 9th International Conference of Intelligent Tutoring Systems (ITS 08), Montreal, Canada, pp. 65–67 (2008).

(9) Ullrich, C. and Melis, E.: “Pedagogically Founded Courseware Generation based on HTN-planning”, Expert Systems with Applications, Vol. 36, No. 5, pp. 9319–9332 (2009).

(10) Karampiperis, P.: “Automatic Learning Object Selection and Sequencing in Web-Based Intelligent Learning Systems Using Learning Technology Specifications”, in Web-Based Intelligent E-Learning Systems: Technologies and Applications, ed. Zongmin, M., Idea Group (2005).

(11) De-Marcos, L., Martinez, J. J., Gutiérrez, J.A., Barchino, R. and Gutiérrez, J. M.: “A New Sequencing Method in Web-Based Education”, Proc. of the Eleventh Conference on Evolutionary Computation (CEC 2009), Trondheim, Norway, pp. 3219–3225 (2009).

(12) Nkambou, R., Frasson, C., Gauthier, G. and Rouane, K.: “An Authoring Model and Tools for Knowledge Engineering in Intelligent Tutoring”, J. of Interactive Learning Research, Vol. 12, No. 4, pp. 323–357 (2001).

(13) Ritter, S. and Blessing, S. B.: “Authoring Tools for Component-based Learning Environments”, J. of the Learning Sciences, Vol. 7, No. 1, pp. 107–131 (1998).

Jaemu Lee received the Ph.D. degree from Hongik University, Korea in 1994. He has been a professor in the Computer Education Department at Busan National University of Education, Korea, since 1987. His research interests include educational ontology, adaptive learning systems, courseware development and instruction methods for the computer.

Yusuke Hayashi received the Ph.D. degree from the Graduate School of Engineering Science, Osaka University, Japan, in 2003. He is an associate professor in the Department of Information Engineering, Graduate School of Engineering, Hiroshima University. His research interests include ontological engineering, instructional design and intelligent learning support systems.

Riichiro Mizoguchi received the Ph.D. degree from the Graduate School of Osaka University in 1977. He was a professor in the Institute of Scientific and Industrial Research, Osaka University until Sept. 20, 2012. He is currently a professor of the Japan Advanced Institute of Science and Technology. His research interests include ontological engineering and intelligent learning support systems.