A Conceptual-level Design Environment for Learning Contents

Yusuke Hayashi

The Institute of Scientific and Industrial Research, Osaka University
8-1, Mihogaoka, Ibaraki, Osaka, 5670047, Japan

hayashi@ei.sanken.osaka-u.ac.jp

Abstract: To meet the needs for large-scale, high-quality learning contents, needless to say, we have to sharpen authoring tools. Authoring process can be roughly divided into two phases, a composing phase and a verification phase. A great deal of effort has been made on the support in the former phase. What seems to be lacking, however, is that in the latter. An ontology-aware authoring tool we have been developing has a function called "Conceptual level simulation. " This supports authors in the latter phase by showing the behavior of learning contents not only as a sequence of concrete behavior but also as structured and abstract behavior along the design intention. Ontology lays the foundation for the function by explicating operational and conceptual semantics of a training scenario.

Keywords: design environment, learning contents, educational systems, ontology

1 Introduction

In the research field on educational systems, large-scale, high-quality learning contents are becoming one of the critical needs of technetronic society. To meet the needs, many researchers address this issue from a variety of viewpoints (Murray, 1999) because the quality of the learning contents depends not only on the author's ability but also on authoring tool's performance.

Authoring process can be roughly divided into two phases, a composing phase and a verification phase. Existing authoring tools support both phases of authors' work to a some extent. However, compared with the support performance for the former phase, one for the latter does not seem very helpful to the author. A typical support function for the latter is to provide a behavior-level test bed where authors can examine their learning contents step by step along the control structure. Of course, it is helpful and absolutely necessary but not very helpful to resolve the logical drawback of learning contents. On analogy of programming, it could be described as "semantic-error debugging." Shapiro explains the hardness of debugging (Shapiro, 1982):

> *A program is a collection of assumptions, which can be arbitrarily complex; its behavior is a consequence of these assumptions; therefore, we cannot, in general, anticipate all the possible behaviors of a give program.*

This is true in case of authoring of learning contents as well. The key to lightening the hardness of debugging is to shift the load to maintain the design assumptions from authors to authoring tools. To realize this, new functions of an authoring tool to be developed include

- A framework for authors to describe design assumptions including design intention of learning contents.
- A function to show the behavior of learning contents not only as a sequence of concrete behavior but also as structured and abstract behavior along the design intention.

We call the latter as "conceptual-level simulation." Our idea is that the structured information generated based on author's design intention will lighten the author's major debugging load to compare what he/she wants (design intention) with what he/she gets (behavior). An ontology (Mizoguchi et al, 1996) plays an important role to embody the above idea. It provides human friendly vocabulary/concepts for authors to describe the learning contents along with design intention. For the authoring tools, on the other hand, it specifies the operational semantics of the learning contents. This operationality enables the conceptual-level simulation of learning contents. Based on this idea, we have developed an ontology-aware authoring tool SmartTrainer/AT (Ikeda et al, 1999; Jin et al, 1999).

2 Ontology-Aware Authoring Tool

Basically, an ontology is a set of definitions of concepts and relationships and a model is a set of instances of them. Roughly speaking, the role of an ontology is to direct the authors towards the correct model. Our idea is that an ontology-aware authoring tool can help authors to reduce the problems of authoring caused by unintentional error and to improve the quality of the product. Our research on SmartTrainer/AT is an embodiment of this idea. We have developed a training task ontology and incorporated it into SmartTrainer/AT as fundamental knowledge source to yield the intelligent functions to support the authoring process.

2.1 Composing learning contents

The authors' task is to write a "training scenario" for SmartTrainer that is a training system engine we have developed. At the appropriate phase of authoring process, the author is required to clarify his/her own idea from the three fundamental viewpoints listed below.

- What type of learner the scenario of the learning content is designed for?
- What educational effect the learning content is supposed to bring about?
- How to achieve it?

Figure 1 shows how the idea (A) is embodied in the learning content (C). The model (B) can be regarded as a representation of design process. An ontology provides vocabulary and concepts, axioms necessary to describe the model. Firstly, an author describes the idea (A) clearly as a topmost, abstract and instructional goal. Then he/she repeats the expansion of the super-goal into relatively concrete sub-goals until a sequence of the sufficiently concrete goals (B-1) is specified. Secondly, he/she designs a sequence of teaching actions (B-2) that are expected to attain the goals (B-1). Thirdly, he/she embodies the actions (B-2) in a sequence of conceptual specification of cards (B-3).

In (B-1) there are two kinds of instructional goals: a goal for diagnosis (D-goal) and a goal for teaching/learning (T/L-goal). A D-goal is a goal to identify the state of a learner. A T/L-goal is a goal to make educational effects on learners. Thus, the author can clarify the first two of the three viewpoints discussed above while describing the model (B-1). At the bottom two levels (B-2) and (B-3), the teaching scenario is characterized from the third viewpoint: "how to achieve it." When the author is very active in referring the ontology, the rationality of the training scenario designed is expected to be quite high. In addition, reusability and sharability of training scenario is also expected to be high, as we have discussed, because an ontology-aware authoring tool stores not only concrete learning content (C) but also

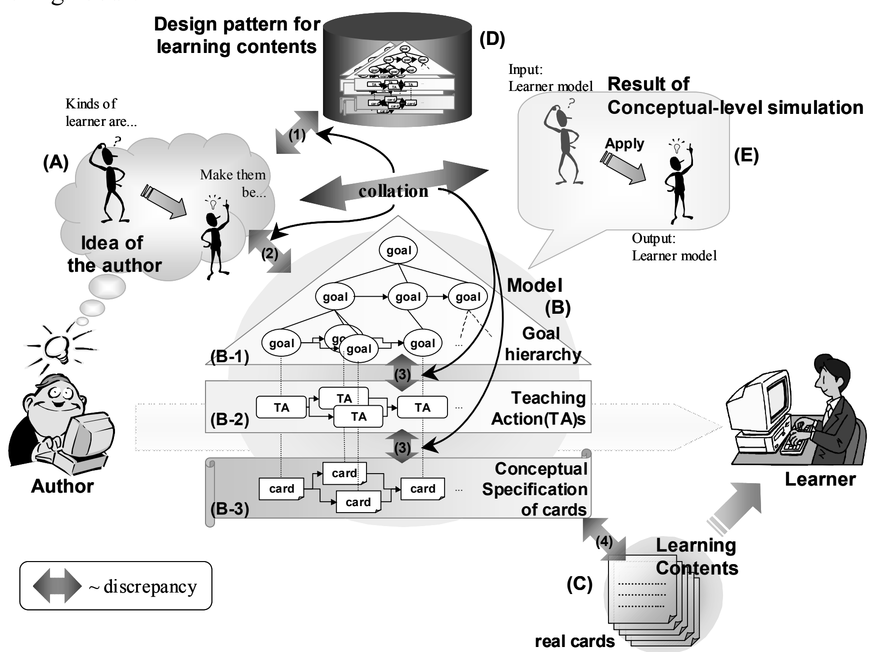

Figure 1: An overview of authoring process

the design intentions and rationale behind it as a model (B) based on the ontology.

2.2 Verifying learning contents

As we have seen in the previous section, an ontology-aware authoring tool is expected to be able to bridge the potential conceptual gap between ideas (A) and presentations (C) and to suppress the unintended error caused by the gap. Models (B) play the important role as a pivot between ideas (A) and presentation (C): as conceptual representation of ideas or as the design intention behind presentation. Needless to say, the error cannot be suppressed completely and there happen to be discrepancies (1)..(4) during design process as shown in Figure 1.

To resolve the discrepancies and be close to perfection, it is very important to identify the location of the discrepancies that are not realized by the author during the design process. An ontology-aware authoring tool provides the information about the discrepancies between what an author wants and what he/she gets. It can be a good cue to identify the discrepancies. The function of our ontology-aware authoring tool is called "conceptual level simulation" which shows the behavior of the training scenario. Conceptual-level simulation can demonstrate the behavior of the training scenario from various viewpoints, along the structure of the design model and may expose the three categories of problems caused by discrepancies (1)..(3) to the author's eye. In the case of forth one, it is rather helpless to resolve the problem caused by discrepancy (4) because conceptual-level model is too abstract to evaluate the quality of real contents.

3 Conceptual-level simulation

In an ontology-aware authoring tool, the design intention remains in an operational form. This means that tools and authors can interpret the behavior of the product from the common viewpoint and enables the tools to provide useful information for authors to interpret it and identify the problems of design. We call the model with operational form of design intention as a conceptual-level model. "Conceptual-level simulation" is a function that simulates the behavior of the conceptual-level model in various levels of abstraction.

As shown in Figure 1(E), the conceptual-level simulation shows the behavior of a training scenario as the change of a learner model. We call the learner model which is turned to an input and an output of the conceptual-level simulation as a "pseudo-learner." In other words, it is a kind of personification of a stereotyped learner in author's mind while he/she is authoring the training scenario. Of course, it is very different from the real learner because we assume its stable and non-autonomous learning behavior. In addition, the pseudo-learners' understanding does not depend on the quality of concrete contents in learning content. This means that a pseudo-learner always succeed in learning what a training system teaches as long as the teaching activities are reasonable from educational principle prescribed in training task ontology.

Figure 2 explains the role of the pseudo-learner in training scenario verifying. It is an ideal situation but almost impossible for authors to be able to examine whether the learning content has the intended educational effect on all the real learners as shown in Figure 2 (A). In Figure 2 (B), by observing the changes taken place in pseudo-learners instead of the real learners, the author can examine whether the conceptual-level model of the learning content is reasonably designed or not.

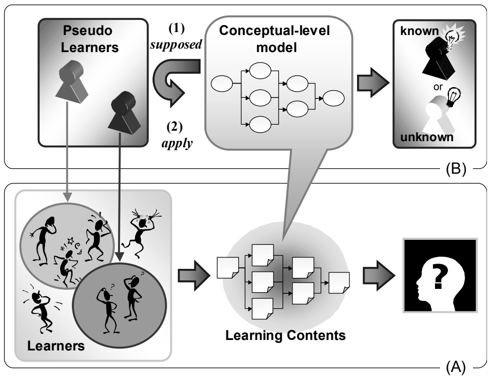

Figure 2: The role of the pseudo-learner in conceptual level simulation

The purpose of the conceptual-level simulation is to show "which part of the training scenario" adds "what type of an educational effect" to "what type of a learner" systematically. D-goals and T/L-goals play the important role to realize the purposeIn the following, we will see how the authoring tool interprets the training model and simulates its behavior briefly.

Classification of learners. D-goals are largely concerned with characterization of a learner in terms of understanding status, grade and ability. Figure 3 shows a correspondence of a structure of D-goals to the classifications of learner. Learners are classified into four kinds of pseudo learner L1,.., L4. This enables ontology-aware authoring tools to provide authors with basic information to verify whether the pseudo learner is appropriately characterized in the training scenario.

Effects on learners. Educational effect of T/L-goal and the necessary conditions to achieve the T/L-goal are specified as abstract axioms in the training task ontology. After an author specified the T/L-goal as a component of the model, the conceptual-level simulator can simulate the effect of the T/L-goal. Intuitively, it adds the educational effects of the T/L-goal to pseudo-learner's status if the necessarily condition of the T/L-goal is satisfied. In Figure 3, the four T/L-goals T1,.., T4 represented by white rectangles are defined. If the training model is well designed, the all the pseudo-learner will understand knowledge units K1 and K2 at the end of the training scenario as shown in Figure 3.

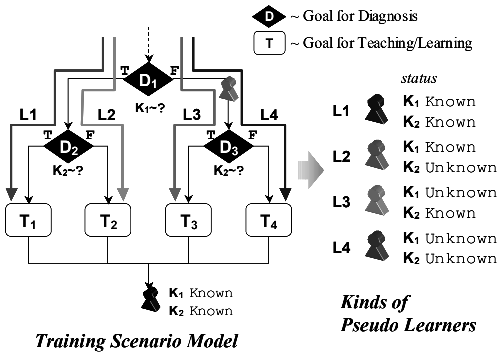

Figure 3: Goal structure and kinds of pseudo-learner

According to these information, an author ascertains whether the training scenario has any problems or not by observing the change of the pseudo learner's status. If some problems are found out, the author has to locate the problems in more detail by getting down into specifics along both the goal structure of training scenario model and the classification structure of the pseudo learners.

4 Conclusion

An ontology-aware authoring tool has functions to effectively reduce the number of the problems which arise during the course of learning contents design. However, there still may remain some problems passed over in the design phase. The conceptual-level simulation helps authors to resolve those problems in verification phase before the training scenario is implemented to the real learner. It simulates the behavior of the conceptual-level model of learning contents in various levels of abstraction and provides good cue to identify the problems.

Having operational semantics specified by the training task ontology enables the conceptual-level simulation of learning contents. The most important role of the ontology is to maintain the continuity from authors' conceptual understanding of learning contents to the operational semantics of them. The implication of "ontology-awareness" and "operationality of the learning contents" is deep. To arrange the best collaboration between authors and tools, it is quite important to create an environment for authors to describe the model easily and for tools to operate the model systematically. One of the most important merits introduced by ontological engineering is that an ontology enables human to share the model with computers. From now on, I intend to consider conceptual-level simulation in detail and to develop an authoring tool with it. Much effort is devoted to standardization of learning content metadata to attain high inter-operability among learning management systems. Conceptual model we have discussed in this paper could be a kind of learning content metadata. Our future plan includes the extension of ontology-aware authoring tool to a comprehensive framework that supports full life cycle of learning contents, that is, design, development, implementation, management and evaluation.

References

Ikeda, M., Hayashi, Y., Jin, L., Chen, W., Bourdeau, J., Seta, K., & Mizoguchi, R. (1999), Ontology More Than a Shared Vocabulary; *Proc. of Workshop on "Ontologies for Intelligent Educational Systems"*, AIED-99, Le Mans, France, pp. 1-10, 1999.

Jin, L., Chen, W., Hayashi, Y., Ikeda, M., Mizoguchi, R., Takaoka, Y. & Ohta, M. (1999), An Ontology-aware Authoring Tool, ~ Functional structure and guidance generation ~, *Proc. of AIED-99*, Le Mans, France, pp. 85~92.

Mizoguchi, R., Sinitsa, K., & Ikeda, M. (1996), Task Ontology Design for Intelligent Educational/Training Systems, *Proc. of Workshop on "Architectures and Methods for Designing Cost-Effective and Reusable ITSs"*, ITS'96, Montreal, pp. 1-21.

Murray, T. (1999), Authoring Intelligent Tutoring Systems: An analysis of the state of the art, *International J. of Artificial Intelligence in Education*, Vol. 10, pp. 98-129.

Shapiro, E. (1982), *Algorithmic Program Debugging*, The MIT Press.