A Multiple View Authoring Tool for Modeling Training Materials

Yusuke HAYASHI, Lai JIN, Kazuhisa SETA, Mitsuru IKEDA, Riichiro MIZOGUCHI
The Institute of Scientific and Industrial Research, Osaka University
8-1, Mihogaoka, Ibaraki, Osaka, 5670047,Japan
Tel. +81-6-6879-8416, Fax. +81-6-6879-2123
{hayashi,kin,seta,ikeda,miz}@ei.sanken.osaka-u.ac.jp

Mamoru OHTA,Yoshiyuki TAKAOKA
TOKO SEIKI Company
83-14-40, Senrioka, Settsu, Osaka, 5660001,Japan
Tel. +81-6-6387-1184, Fax. +81-6-6389-7032
{ohta,charly}@uno.cdd.toko-s.co.jp

ABSTRACT: This paper discusses ontology-aware authoring tool for training systems. This tool uses the ontology as the source of its intelligence. The ontology plays the role of a meta-model which helps authors to construct a model. The major advantages of such an authoring tool are: (1) Providing human-friendly primitives (descriptiveness, readability). (2) Providing multiple-views of the single model of training. (3) Simulating the abstract behavior of the model (conceptual level operationality). (4) Verifying the consistency of models. In this paper we discuss the roles of ontology in authoring tools focusing on (1), (2) and (3). In addition we exemplify the benefits of ontology by giving an overview of the authoring tool SmartTrainer/AT.

KEYWORD: Ontology in Education, Contents Production with Authoring Systems, Industry Training System

1. Introduction

Building an intelligent educational system (IES) requires a great deal of work. Currently, however, such systems are always built from scratch. Few functional components are reusable and existing systems are hard to compared or assessed. The only existing contributions to the solution of this problem are found in the study of authoring tools for educational systems. However, it is considered questionable whether such tools provide substantial benefits for the authors during the course of IES construction, since most existing authoring tools do not satisfy the requirements shown below.

- To provide human-friendly primitives which authors can easily use to describe their own idea of an IES.
- To give appropriate guidance to authors based on the established principles of the educational task.
- To show the dynamic behavior of the IES at a conceptual level for examination of its validity.

We believe that the key to the solution to this problem is intelligent support based on the principle of “ontology”, which serves as a theory of vocabulary/concepts used as building blocks for human centric software design.

Recent efforts [NKA 96, MUR 96, RED 97, IKE 97, DUB 97, MUR 98, JIN 99] have been directed toward designing IES-Authoring tools that meet the requirements specifies above. Eon tool [MUR98], in particular, is a well-designed authoring tool for knowledge-based tutors with customizable ontology. Authors can define the types of topics, topic links, and topic levels, for the topic ontology and can also define the ontology of allowable values for the student model. His ontology, however, is vocabulary and no more. In [MUR98] Murray pointed out “the field needs more in the way common terminologies for describing domain and teaching knowledge to better compare systems and share knowledge base”. This seems to be along the same lines as our ontological research roadmap [MIZ 96, MIZ 97, IKE 97].

Our tool is said to be “ontology-aware” in the sense that it understands the ontology that is the source of its intelligence. The ontology plays the role of a meta-model that helps authors build a model, that is, a training system in a certain domain in our case. This is not a hard-wired model but a set of building blocks with formal semantics, therefore the tool is both flexible and intelligent. This approach makes the tool user-friendly and helpful, since it uses vocabulary at the lexical level at which authors communicate with it, and the axioms or semantic constraints associated with the ontologies can generate timely messages of appropriate types when they are violated. In addition it clarifies the relationship between views of models since it can manage identity and changes of models with the axioms or semantic constraints. In addition, our ontology aware-authoring tool can provide multiple-views of the single model of training. Each of the views is so designed

```
1  (V: Teaching_Material_Structure
2      v/o+ AbstructDesign::
               AbstructDesign)
3      v/o+ AbstructStructure:
               S:Teaching_Action)
4      v/o+ ConcreteStructure: S:Action).
5    -(V: Courseware_Structure
6    -(V: Backbornstream_Structure
7    -(V: Ribstream_Structure
8  (Goal
9      p/o+ Target:Learner)
10     p/o@n Topic: Learning_Item)
11     p/o+ Before: Knowledge_State)
12     p/o+ After: Knowledge_State)
13     p/o@n Subgoal: Goal).
14 (Learning_Item
15     -(Composite_ Learning_Item
16       p/o+ Label: Text
17       p/o@n Element: Learning_Item).
18     -(Relation_ Learning_Item
19     -(Single_ Learning_Item
20 (Re: Learning_Item_Link
21     p/o@2 learning_Item: learning_Item)
22   -(Re: Ex-Rule_Link
23       p/o Rule: learning_Item)
24       p/o Example: learning_Item).
25   -(Re: Generalize_Link
26       p/o General: learning_Item)
27       p/o Special: learning_Item).
28   -(Re: Part-Whole_Link
29       p/o Whole: learning_Item)
30       p/o Part: learning_Item).
31 (Learning_Item_Network
32     p/o@n Node: Learning_Item)
33     p/o@n Link: Learning_Item_Link).
34 (Learner_Model
35     p/o+ Target_Knoledege:
           Learning_Item_Network)
36     p/o@n Knowledge_State:
             Knowledge_State).
37 (Teaching_Activity
38     p/o+ Topic: Learning_Item)
39     Rep-: Goall),
40   -(Ask_a_Question
41       p/o+ Topic: Learning_Item).
42   -(Give_an_Explanation
43       p/o+ Topic: Learning_Item).
44   -(Simulate_Operation
45       p/o+ Topic: Learning_Item).
46   -(Give_a_Hint
47       p/o+ Topic: Learning_Item).
48 (Action
49     Rep-: Teaching_Activity).
50   -(Show_a_Card
51       p/o+ Teaching_Material: Card)
52       Rep-: Give_an_Explanation)
               | Simulate_Operation).
53   -(Show_a_Dialog
54       p/o+ Teaching_Material: Dialog)
55       Rep-: Give_a_Hint).
56   -(Highlight_a_Point
57       p/o+ Notable_Point: Item)
58       Rep-: Give_a_Hint).
59 (Problem_List
60     s/o@n Problem: Teaching_Action)
61 (Ro: Problem%Problem_List
62     p/o+ Topic: Learning_Item)
63     p/o@n Diagnosis: Diagnosis)
64     p/o+ ObjectScope: Unit)
```

Note

Concept:

"(":concept definition, ")":concept reference

"B:": Basic concept(omission), "Re:": Relation concept, "A: ": auxiliary concept, "Ro: ": Role concept, "Ct: ": Category concept, "S: ": Status concept, "V: ": View concept, "$:": Set concept, "S:": Sequence concept, "Text": Annotation

Relationship:

"p/o+":part-of, "s/o":sequence-of, "a/o":attribute-of, "v/o":view-of, "p/i":participate-in, "Rep-:":representation, "+":the singlular, "@n":the plural

"x:y":"x" represents the slot name, "y" represents the classconstraint

Is-a: Describe the relationships between general concept and sepecific concept.

part-of: Describe the relationships between whole and part. attribute-of: Means in the slot is the value of attribute.

figure 1 Training task ontology(A portion)

as to be well adapted to the varying context of authoring process. The consistency among the views is maintained based on the axioms or semantic constraints specified in the ontology.

We could say an ontology-aware authoring tool can share the meta-model/vocabularies with authors and provide authors comfortable workplace to build teaching materials. In this paper, we will exemplify the benefits of ontology using SmartTrainer/AT, an authoring tool developed by the authors.

2. An ontology-aware authoring tool

We have developed a training task ontology for Training systems. Figure 1 shows a portion of the conceptual level of the training task ontology. Task ontology represents problem solving process domain-independently [MIZ 95][SET 97]. Domain ontology represents an object of training. It is integrated with the training task ontology that includes training scenario, learner model and learning item network, and then acquires the appropriate task-dependent meaning in T-Domain ontology. In the substation operator training case, a task is "training" and a domain is "substation in the electric power network". By incorporating the ontologies into the authoring tool, the support functions shown above have been realized in an affective manner.

Basically, an ontology is a set of definitions of concepts and relationship 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 ontology is two layered: conceptual-level ontology and lexical-level ontology. Lexical-level ontology provides a human-friendly vocabulary used by authors for describing their intended model of teaching materials. The vocabulary is so organized in a systematic manner, so that authors can easily find appropriate words to express their thoughts. Conceptual-level ontology specifies the detailed meaning of each concept and the relationship between them. These two level ontologies have correspondency with each other. The correspondency enables to translate one level model into the other level model.

The lexical-level model is a description of the intended model in the author's mind. In general, it is difficult for humans to externalize their ideas accurately. However, it is usually possible to communicate with each other smoothly because of the remarkable human ability of insight into other's intentions. Needless to say, shared knowledge is the source of this insight. Lexical-level ontology, conceptual-level ontology, and the correspondence between them act as the shared knowledge for smooth communication between the author and the tool. The conceptual-level model represents the tool's interpretation of a lexical-level model.

One of the characteristic features of having a conceptual level training task ontology is the implementation of a unique function, "conceptual level execution", which shows the authors the dynamic behavior of the model using a conceptual level representation. The author may examine whether or not the training model behaves well for the learner by monitoring the result of the conceptual level execution, before teaching materials are finalized. In addition, the ontology also allows us to give us appropriate guidance in the authoring tool [JIN 99].

However a rigorous representation at the conceptual level ontology is necessary for the authoring tool to provide an intelligent support functionality. This is not always available if the teaching material author has a low level of ontology literacy. A typical example of a difficult ontological distinction is between an "inherent" concept and a "role" concept. Each concept in the training task ontology can be defined as either an "inherent" concept or a "role" concept. For example, when a "Learner" has two meaning, that is, one as a basic concept of a "human" and another as a role concept of "a person who learns something". The distinction among the types of the concepts, such as "inherent", "role", "status" and "category" is very important when building an operational conceptual model. However, on the other hand, we cannot expect average authors with low ontology literacy level to distinguish between them exactly. Lexical-level ontology provides authors with an appropriate interface suited for his/her ontology literacy.

Therefore we have developed the training task ontology on two levels, the conceptual level and the lexical level as shown above. By incorporating these two levels of ontology into the authoring tools, both user-friendliness of interface and intelligence of active support functions can be realized.

3. SmartTrainer/AT

Figure 2 shows an overview of the whole environment, including an ontology editor, an authoring environment, SmartTrainer/AT, and an intelligent training system SmartTrainer. Note that all of those environments share the single ontology which acts as the source of the environments' intelligence.

The users of the environment may be assigned to one of the three categories, ontology authors, model authors and learners. The environment provides an interface adapted for each category of the authors. An ontology author, who uses the interface shown at the top of figure 2, builds the ontology which is further partitioned into three sub-ontologies, domain ontology, task ontology, and T-domain ontology as discussed previously. Following the ontology, a model author creates a model through the interface, SmartTrainer/AT in our case, shown at the bottom of figure 2. In figure 2, we draw a distinction between domain model authors and teaching material authors to explain their activities clearly. However, the distinction is not necessarily essential since one person can build both models in most cases. A domain model author creates the model of an electric power network by instantiating the device concepts and connecting the instances to each other. If some instantiations or connections are found to be inappropriate according to the domain ontology, SmartTrainer/AT provides the author with a guidance message to lead him/her towards an appropriate model. A task model author creates the teaching materials which are an integrated model of task and domain, called the, T-domain model. The author designs the structure of the intended teaching material, while referring to the domain model. The task ontology provides two types of basic teaching material structure, static and dynamic. Static structure, called a learning item network, specifies various kinds of educational relationships between the learning items, for example, "prerequisite", "easier-than", "example-of" and so on. The dynamic structure, called the presentational structure, specifies the control flow of the training, that is, how and when each learning item should be presented to a learner. The presentational structure has two subtypes of structures, a courseware and a training scenario. Basically, the former describes the systematic method of educational presentation and the latter the opportunistic approach is to support situated-learning.

The learning item network, courseware, and training scenario is explained in more detail in the following sections. Note that all of these environments share a single ontology which allows smooth communication between them.

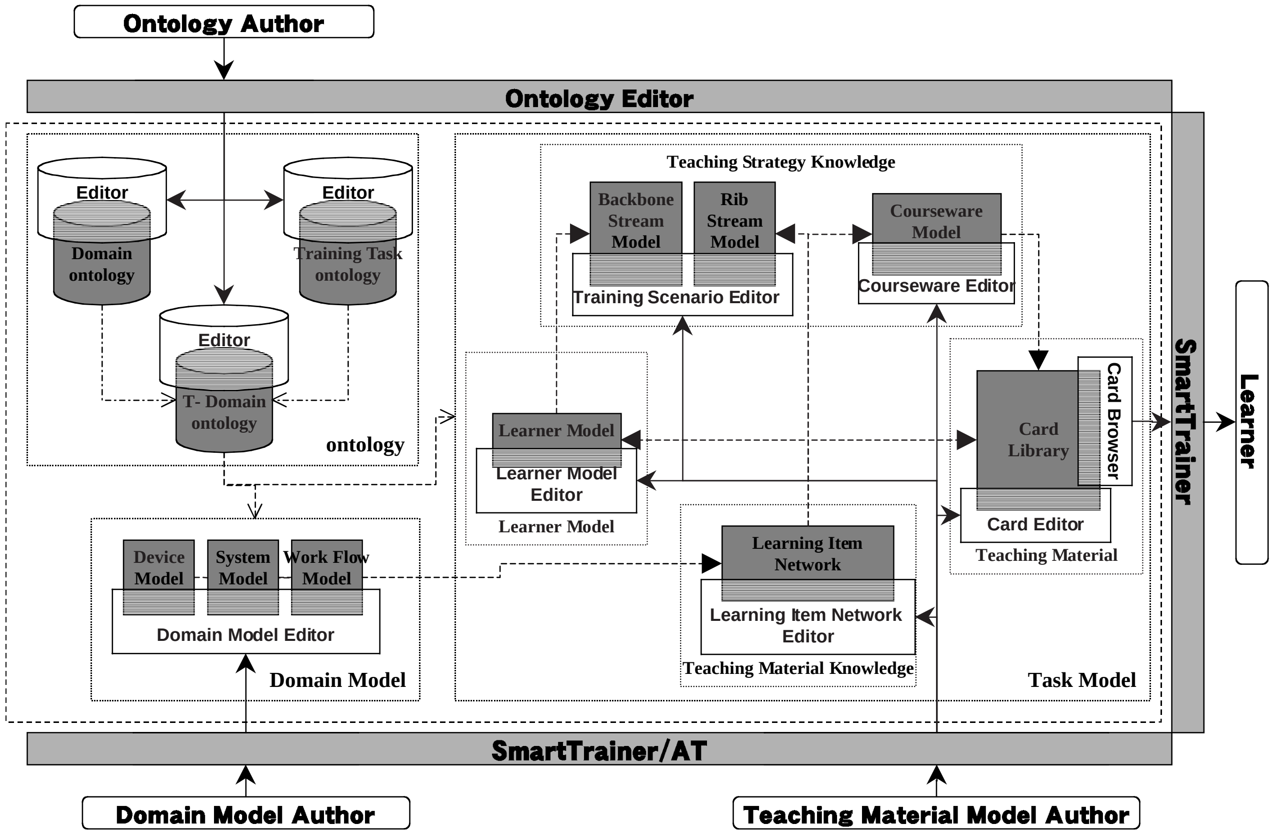

Figure 2 Overview of SmartTrainer/AT

4. Authoring processes in SmartTrainer/AT

Figures 3~6 show snapshots of the interfaces of the authoring tools we have developed. Most of the words that appear on the interfaces are from the lexical level task ontology. By selecting appropriate words from the menu or vocabulary browser authors describe the intended model for the training at the lexical level. The lexical level model is then translated into the conceptual level model that fully specifies the operational semantics. By maintaining semantic consistency of the model, SmartTrainer/AT directs authors to a valid model. The two level configuration of the ontology achieves both readability and operationality of the task model.

4.1 Learning Item Network

A learning item is a source of information used to make a presentational structure. The concepts in domain ontology are given the task role of “Learning item”, at the first step of building teaching materials. The learning item network is structured along with the training task, with relationships between learning items, for example “is-a”, ”part-of”, “prerequisite”, ”harder-than” and “similar-to”, and so on. A learning item is the task concept closest to the domain concept in the training task ontology. Figure 3 shows the window used to edit the learning item network. Window (1) is for browsing/editing the learning item network using a bird’s-eye-view. Window (2) is for browsing/editing the learning item network focusing on the learning item selected in window (1).

An author may examine/edit the contents of each learning item using window (3). The contents of each learning item come from the domain ontology and model. In addition, the author can augment this with additional task-contextual information, including a reference to the appropriate learning cards used to teach the learning item.

Behind the interface, SmartTrainer/AT maintains the consistency of the model according to the conceptual semantics specified by the domain ontology and task ontology. It helps author to keep the teaching material up-to-date with respect to the actual subject. When the domain mode is changed, for example, a device in a certain plant is replaced with a new one, SmartTrainer/AT can recommend that the authors update the related portion of teaching material.

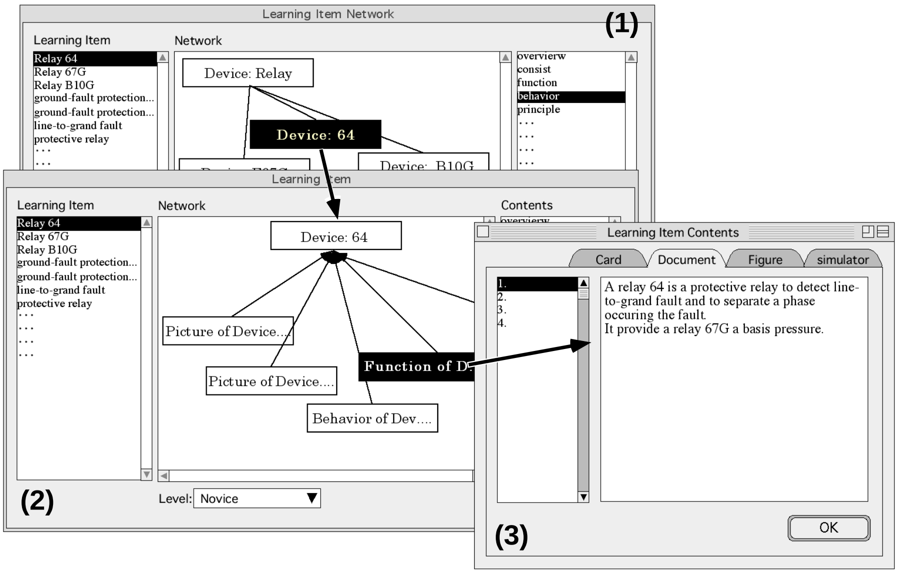

Figure 3 Learning Item Network

4.2 Training Scenarios

A training scenario is a presentational structure by which learners practice the target operation according to workflow. Figure 4 shows the windows used by authors to describe the training scenarios. The training scenario in window (2) consists of a “backbone stream”, which is a sequence of questions along the workflow and a “rib stream”, which is the treatment for learner’s erroneous answers to the backbone questions. Before the window appears, the author is asked to select one of the target workflows in the domain model and partition it into several parts, as shown in window (1). The author then makes a question list for the backbone stream corresponding to the target workflow in window (2).

The author enters the “question intention” and “question” in windows (3) and (4), respectively. “Question intention” is

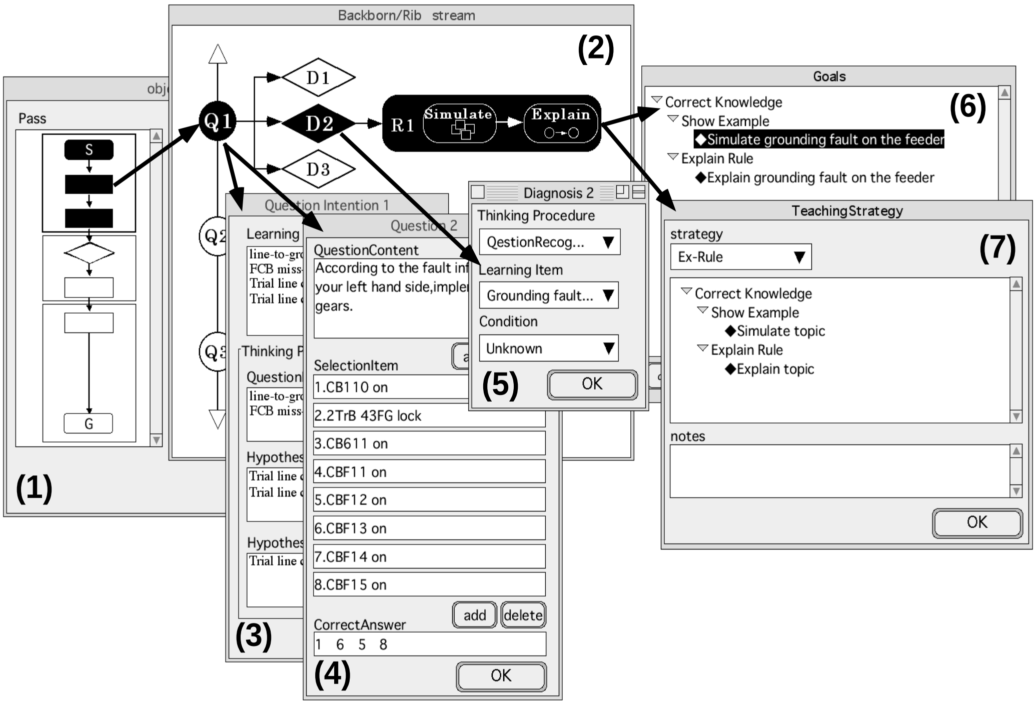

Figure 4 Training Scenarios

a description of the subject of the question, that is, the key domain knowledge required to answer the question. In window (4), the author enters the components of the question/answer, that is, "question", "items to be ordered" and "correct answer".

In window (2), a series of questions, Q1, Q2, and Q3, represent a backbone stream and a series of activities, denoted by R1, represents a ribstream. The diagnosis D2 represents a condition required to execute the rib stream R1. D2 is defined as "The learner makes a mistake because he/she doesn't know the knowledge about a feeder grounding accident" as shown in (5). Once ribstream R1 is activated, it "simulates a topic" and "explains the topic" to the learner in that order. The window (6) shows a goal tree as a design intention behind R1. The top goal, "correct knowledge" of R1, is expanded into the series of subgoals, "give an example" and "explain the rule." This expansion is done based on the teaching strategy, which shows the correct arrangement of teaching activities/subgoals required to achieve the upper goal and act as a design guideline for the presentational structure. Basically, teaching strategies should be derived from educational principles. Some of them, however, may be heuristic and use subjective knowledge. Thus SmartTrainer/AT uses teaching strategies as a weak constraint for proper modeling to provide the authors with suggestions or recommendations. In addition to this, an author can also design his/her own teaching strategies which may be reused and shared with other authors. In figure 4, the teaching strategy "Ex-rule" in window (7) is used to expand the top goal "correct knowledge" in window (6). Thus, the goal of R1 is well described in design phase and provides a good basis for maintenance/reuse of the teaching materials.

As shown in figure 4, the author can describe the model by using vocabulary in a lexical-level ontology. SmartTrainer/AT provides authors with different views of the teaching material through well designed interfaces in order to cope with a variety of authoring contents, for example, (1) for domain modeling, (3), (6) and (7) for abstract presentational structure design and (4) and (6) for teaching activity arrangement.

4.3 Coursewares

Courseware is a presentational structure by which learners acquire the subject domain knowledge in a systematic manner. Figure 5 shows the windows used to build a courseware. SmartTrainer/AT can guide the author smoothly through the process of construction, from abstract presentational structure to concrete one. First, the author specifies a courseware structure that consists of chapters, sections and paragraphs as shown in window (1). Secondly, he/she edits goals corresponding to courseware's structure as shown in window (2). These goals express the educational intentions of each part of the courseware. The author then select an action from several types of actions defined in training task ontology as in

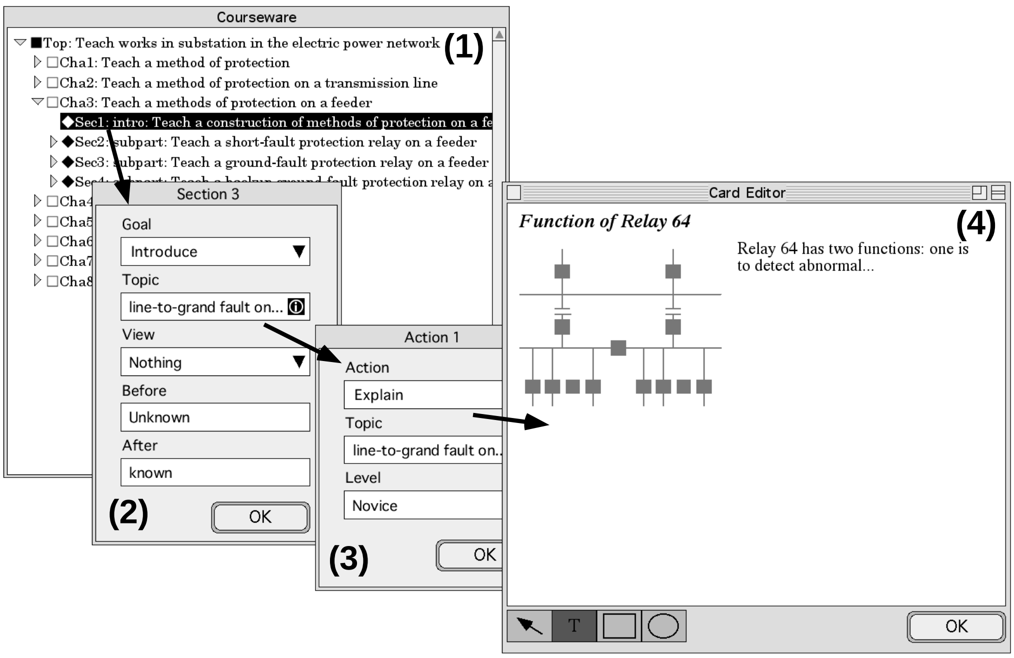

Figure 5 Courseware

window (3). In SmartTrainer, the concrete teaching material represented as hyper-linked cards in which learning items explained in a variety of forms, such as text, graphics, sounds and so on. At the last step of the authoring process, as shown in window (4), he/she creates a flow of cards as a concrete courseware.

4.4 Conceptual level execution

Having a detailed conceptual model enables an ontology-aware tool to provide intelligent support functions for authors. A typical function of this type is the conceptual execution of the created presentation. As the presentation control structure, the learning item network, and a set of cards and the relationships between them are stored in the conceptual-level T-Domain model, the ontology-aware tool can simulate any change in the learner's understanding according to the presentation control structure. This simulation is known as "conceptual level execution."

Once a conceptual level model has been generated, SmartTrainer/AT can execute it based on axioms about a change of models by processes described in the task ontology. For example the effect of the process "Explain concept C to the learner L", is defined as "If the learner L is able to achieve the learning process, he/she can understand concept C."

The author can examine whether or not the training model behaves well for the learner by monitoring the results of the conceptual level execution. Figure 6 shows a imaginary snapshot of the interface used to monitor conceptual level execution. Window (2) is used to set parameters in the learner model and target teaching material. A learner model is defined as a simple overlay model on the learning item network. In this window the author usually edits a property of a simulated-learner and selects target teaching material.

When the author executes it, the executor changes the color of the path along which the learner will learn as shown in window (1), and shows the process of changes in the learner model by changing the color of each node as shown at the bottom part of window (2). Using this function the author can verify that teaching material appropriately reflects the intention of the author and identify any insufficient section that may exist.

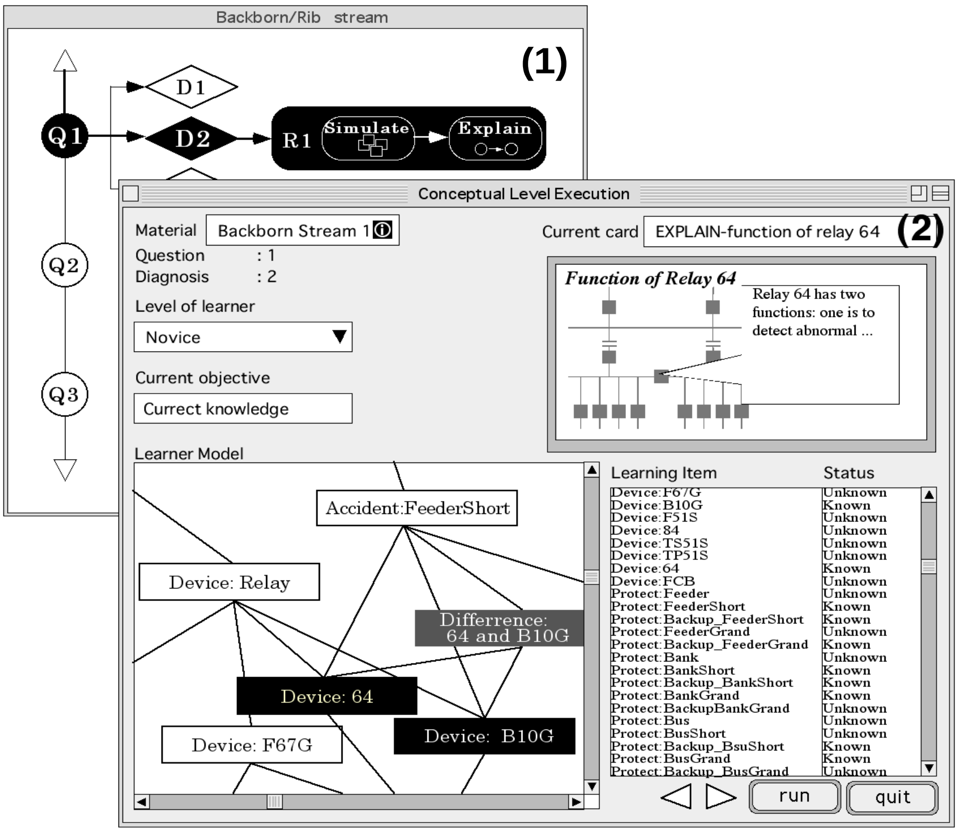

Figure 6 Conceptual level execution

5. Concluding remarks

In this paper we have discussed the roles of ontology to implement easy-to-use interfaces of SmartTrainer/AT. Two levels of the ontology, lexical and conceptual, make the authoring tool user-friendly and helpful. Lexical level ontology provides authors with an interface suited to his/her ontology literacy by using appropriate vocabulary. Simultaneously the conceptual level ontology specifies the conceptual semantics of the vocabulary and directs the authors towards the correct model using techniques such as “conceptual level execution”. Furthermore, SmartTrainer/AT can provide Multiple-views of teaching materials, because the consistency among the views is maintained based on the axioms or semantic constraints specified in task/domain ontology. So each of the views is so designed as to be well adapted to the varying context of authoring process. These function in SmartTrainer/AT provide authors comfortable workplace to build teaching materials.

We have exemplified the benefits of ontology using SmartTrainer/AT. We have been developing SmartTrainer/AT using JAVA and CLOS. Domain experts comment that SmartTrainer/AT provides a well-designed suite of tools. We intend to make SmartTrainer/AT more user-friendly and intelligent in future work. For example we have been developing a framework of teaching strategies as patterns to design teaching materials. In the object-oriented paradigm, design patterns [GAM 94] are becoming popular recently. They are obtained by abstracting the rich experiences of object-oriented modeling and can provide suggestions useful during the course of model building. Our idea is to represent patterns to design teaching materials in terms of an ontology and hence they are operational. An authoring tool can understand them and provide the authors with reasonable suggestions based on them.

Reference:

[DUB 97] duBoulay, B. Proc. of the 8th World Conf. on AI-ED, pp. 565-567, Amsterdam: IOS Press.

[GAM 94] Gamma, E., Helm, R. Johnson andVissides. J.: Design Paterns: elements of reusable object-oriented software, Addison-Wesley Publishing Company, 1994.

[IKE 97] Ikeda, M., Seta, K., Mizoguchi, R.: Task Ontology Makes It Easier To Use Authoring Tools, IJCAI-97, Nagoya Japan, pp.342~347, 1997.

[JIN 99] Jin, L., Chen, W., Hayashi, Y., Ikeda, M., Mizoguchi, R., Takaoka, Y., Ohta, M.: An Ontology-Aware Authoring Tool ~ Functional structure and guidance generation ~, Proc. of AI-ED 99, Le Mans, (to appear), 1999

[MIZ 95] Mizoguchi, R., Vanwelkenhuysen, J. and Ikeda, M.: Task Ontology for Reuse of Problem Solving Knowledge, Knowledge Building & Knowledge Sharing 1995 (KB&KS’95) (2nd International Conference on Very Large-Scale Knowledge Bases), Enschede, The Netherlands, pp.46-59, 1995.

[MIZ 96] Mizoguchi, R. et. al. Task Ontology design for intelligent educational/training systems, Workshop on Architectures and methods for Designing Cost-effective and Reusable ITSs, ITS96, Montoreal.

[MIZ 97] Mizoguchi, R., Ikeda, M., and K. Sinitsa. Roles of Shared Ontology in AI-ED Research, —Intelligence, Conceptualization, Standardization, and Reusability —. Proc. of AIED-97, Kobe, Japan, pp. 537-544 (1997).

[MUR 96] Murray, T.: Special Purpose Ontologies and the Representation of Pedagogical Knowledge, ICLS96.

[MUR 98] Murray, T: Authoring Knowledge-Based Tutors: Tools for Content, Instructional, Strategy, Student Model, and Interface Design; The journal of the learning sciences, 7(1), pp. 5~64, 1998.

[NKA 96] Nkamkbou, R. Gauthier, R. Frasson, C. CREAM-tools: An authoring environment for curriculum and course building in an ITS. Springer-Verlag: Proc. of the 3rd Int’l. Conf. on CAL& Inst’l Sc. & Eng.

[RED 97] Redfield, C. L: An ITS Authoring Tool: Experimental Advanced Instructional Design Advisor. In papers from the 1997 AAAI fall symposium (Tech. Rep. No. FS-97-01, pp. 72-78) Menlo Park, CA: AAAI

[SET 97] Seta, K., Ikeda, M., Kakusho, O. and Mizoguchi, R.: Capturing a Conceptual Model for End-user Programming -Task Ontology as a Static User Model-, Proc. of the sixth international conference on user modeling,UM’97, pp. 203-214, 1997.