Abstract

This paper discusses ontology-awareness of authoring tools. The most important role of ontology is to lay the theoretical foundation for IES development process. It maintains continuity from author’s conceptual understanding of educational task to the computational semantics of IESs. It provides human friendly vocabulary for authors to describe the educational task. For the authoring tools, on the other hand, it specifies the computational semantics of vocabulary. The goals of our research on ontology are to exemplify the benefits of ontology more than vocabularies through the development of an ontology-aware authoring tool for intelligent educational systems.

1. Introduction

Most of existing commercial authoring tools with sophisticated user interfaces are very easy to use, because it is designed based on the deep understanding of authors’ demands. Certainly, it is one form of ideal human-centric software design. However, the authors sometimes do not feel easy when using the tools, because they do not understand author’s intention very well. For example, they might be upset by an absurd restriction, if they could not appreciate the significance of the restriction. This implies that the authors are satisfied with the tools at surface level and not at deep level. The restrictions, however, should be reasonably significant if they are devised by the tool designer who deeply understands the author’s mind and educational common sense. The problem mostly comes from two kinds of ignorance. One is the authoring tools’ ignorance of the designer’s mind, and the other is the designers’ ignorance of the model of educational systems.

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. This is not a hard-wired model but a set of building blocks with formal semantics, therefore the tool is both flexible and can intelligent. This approach makes the tool user-friendly and helpful, since it uses vocabulary at the conceptual level which authors can use to communicate with it, and the axioms or semantic constraints, associated with the ontologies can generate timely messages of appropriate types when they are violated. We could say an ontology-aware authoring tool can share the meta-model/vocabularies with authors and then the two kinds of ignorance above would go away. In this paper, we will discuss basic issues about ontology-aware authoring tools.

2. An ontology-aware authoring tool

Building an IES requires a great deal of work. Currently, however, such systems are always built from scratch. Few functional components are reusable and existing systems con not be compared or accessed. 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 authors engaged in construction of an IES, since most existing authoring tools do not satisfy the requirements as shown below.

- To provide human-friendly primitives, which authors can easily use to describe their own skeleton IES.

- To give appropriate guidance to authors, based on the established principles of the educational task, by checking the rationality of the skeleton IES.

- To show the dynamic behavior of the IES at a conceptual level, to enable the authors to examine its validity.

We believe that the key to the solution of 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 [NKA96, MUR96, RED97, IKE97, MUR98, JIN99] in the AIED community have been directed toward designing ITS-Authoring tools that meet the requirements specified 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. The 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 [MIZ96, MIZ97, IKE97].

The goals of our research on ontology are to exemplify the benefits of ontology more than vocabularies through the development of an ontology-aware authoring tool for intelligent educational systems. In the following, we will explain basic ideas of “ontology-aware” authoring tool along the figure 1.

2.1 Basic structure: ontologies and models

The basic structure of ontologies and models are shown in figure 1 (a). 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 toward the correct model. Fundamentally, ontology is composed of two parts, that is, conceptual-level ontology and lexical-level ontology. Lexical-level ontology provides a human-friendly vocabulary of terms used by the authors to describe the intended model of teaching materials. The vocabulary is so organized in systematic manner so that authors may easily reach appropriate words to express their thoughts. Conceptual-level ontology specifies the detailed meaning of each concept and relationship between concepts. Correspondence between the two ontology lays the foundation for the translation between a lexical-level model and a conceptual-level one.

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 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 tools. The conceptual-level model represents the tool’s interpretation of a lexical-level model.

2.2 Integration of task ontology and domain ontology

We have employed two kinds of ontologies: One is “task ontology”[Mizoguchi, 1995;
KBKS1995][Seta; 1997] which is one for representing problem solving process domain-independently and the other is “domain ontology” which corresponds to ordinary one. In the substation operator training case, a task is “training” and a domain is “substation in the electric power network”. Those are shown at the bottom of figure 1. Training task ontology provides us with unit activities and task-dependent concepts which collectively specify the appropriate context where each domain concept should be put.

Domain ontology in case of the substation operator training consists of “device”, “electric network” and “workflow”. It is integrated with the training task ontology which includes
training scenario, learner model and learning item network, and then acquires the appropriate task-dependent meaning in T-Domain ontology.

The two ontologies are integrated into T-Domain ontology shown at the center of figure 1. A typical T-Domain concept is a learning topic/item. It has the meaning of the corresponding domain concept on one hand, and the task-context meaning on the other hand. T-D binding integrates a task concept and a domain concept into a T-Domain concept as shown in figure 2 (a), where gray objects represent task concepts and the white circles represent domain concepts. The domain ontology in figure 2 includes two domain concepts of a “64G” and a “Relay”, and a conceptual relation is-a between them. The two domain concepts are integrated with the task concept of learning item by T-D binding schemes. The T-Domain concepts of learning item inherit the meaning of contents from domain ontology and the task contextual semantics from task ontology.

The task semantics of the activity “teach-prerequisite-topic”, for example, specifies that “PREREQUISITE” relation should hold between the target topic and the current topic of the activity. The two learning items shown in figure 2 meet the specification.

In addition to such static semantics, task ontology specifies dynamic semantics of the activity as shown in figure 3. It shows how TEACH-TOPIC activity is expected to act upon the learner’s understanding. An overlay model, which represent learner’s understanding as a set of understanding statuses of the learning items, is adopted as the learner model representation. The effect of the activity is represented by change of understanding status of the learning items. The change is propagated to the learner model and then causes version change to the learner model. Note that the learning item has empty circle inside. It implies that the dynamic semantics is specified domain independently.

The separation between task and domain enables the reuse of the knowledge. For example, not only training but also a variety of tasks such as facility maintenance, trouble shooting, and personnel affairs, should be coped with by information systems for enterprise knowledge management. The separation between task and domain enable the single domain model to be used for the different tasks.

2.3 Overview of authoring process

2.3.1 Authoring of domain model

An author firstly builds a model of the electric substation operation with domain model editor as shown at the right bottom of figure 1. He/she, for example, creates the instances of devices, names them, and connects them up with each other. Lexical-level domain ontology provides the author with the domain vocabulary through the domain-model editor, while conceptual-level one leads him/her into the proper domain modeling by giving the definitional constraint over the domain concepts.
The role of domain model editor is not only to provide an easy-to-use graphical interface with authors, but also to make two-way translations between lexical-level and the conceptual-level. The translation from lexical-level to conceptual level corresponds to making the interpretation of author’s description. On the other hand, information created in conceptual level is translated into lexical-level statements for the author to understand it easily. For example, a violation of definitional constraints, which can be detected only at conceptual-level, is translated into lexical-level statements and then is notified to the author as a guidance message. The translation mechanism is very important to fill the literacy gap between authors and ontology-aware authoring tools.

2.3.2 Authoring of T-Domain Model

A domain model lays a basic structure of the learning item network. A Learning item network represents the educational organization of domain concepts as we have discussed in 2.2. Authors work is to establish educational relations among learning items and augment them with educational attributes such as “difficult” or “easy”. A learning item network represents a declarative educational organization of teaching material, while courseware and training scenario represent a presentational organization of it. The presentational organization basically specifies the control flow of the teaching material, that is, “when or how learning items are best presented to learners.” Author’s work here is to arrange the learning items into the presentational structure. Ideally, the structure is designed properly based on instruction design theory. We will discuss this issue later in 2.5 and 3.

Courseware presentation is arranged in systematic order of learning in principle, for example, from easy to difficult, from simple to complex, or from theory to application. On the other hand, training scenario presentation is arranged along the process of target workflow, for example, electric substation operation in our case. In our training system SmartTrainer, we adopt backbone stream and rib stream as basic presentation structures to embody a kind of situated learning. A backbone stream is a sequence of questions about the learning items included in the accident recovering process which should be mastered by the learner. A rib stream consists of a series of teaching behaviors for instructing each piece of the knowledge corresponding to the question in the backbone stream. When the learner makes a mistake in the backbone stream, an suitable rib stream will be selected to help him/her to learn according to the learner model and the intention behind the question.

Design of the presentation can be reasonable, if some established educational principles, which might be either heuristical or theoretical, support it. Teaching strategy is one form of the principle to arrange the presentation. The teaching strategy specifies how an instructional goal is expanded to a series of subgoals or actions. With the aid of conceptual level representation of teaching strategies, training model editor guides the authoring process based on the strategies, asking authors to state their intended instructional goals behind the presentation step by step.

The presentation control refers the learner model to conduct the training process adaptively to
each learner’s situation. Author needs to put the reference of learner model on the specification of the presentation. Since the learner model is a simple overlay on the learning item network, an author does not have to do a little to define it.

2.3.3 Authoring of Cards

The most primitive presentation entity of SmartTrainer is a “card”. All the educational contents including simulators are arranged on a card in the form of text, figures, sounds, video, and interface widgets. Author creates a set of cards following out the presentation design. Although we have not implemented card editor yet, it will be able to provide ontology-aware support functions for maintaining the card library.

Each card in the library is associated with two portion of the T-Domain model, that is, the learning item network and the presentation structure. The former associations represent what learning contents each card includes and the latter ones represent which presentational context it was used. The set of associations for a card collectively implies “design intention” of a card. Using the design intention, the authoring tool provide helpful functions for authors to reuse the existing cards.

![Diagram of conceptual execution](image)

**Figure 4 Conceptual-level execution**

2.3.4 Conceptual execution of presentation

Having a detailed conceptual model as shown in figure 3 enables an ontology-aware tool to provide intelligent support functions authors. Such a typical function is conceptual execution of the presentation created.

Figure 4 is the imaginary window which shows the result of conceptual level execution of an instructional presentation, since we have not completed yet the implementation of the conceptual execution. As the presentation control structure, the learning item network, and a set of cards and the associations among them in conceptual-level T-Domain model, the ontology-aware tool can simulates the learner’s change in understanding along the presentation control structure. We call the simulation as “conceptual execution.” The executor assumes the learner’s responses to the questions along the training scenario,
activates the teaching strategies and updates the learner model based on the definition of teaching activities defined in task-ontology. Showing the dynamic behavior of the model to the author, the executor can check the validity of the training model based on the task ontology. When some defects or shortcoming of the model are identified, the executor informs the authors and suggests a desirable way to make alternation to the model.

2.3.5 Generation of a target system

After all the presentation structure and a set of cards is created, the target system can be generated. This process shown in upper left of figure 1 can be regarded as compilation in which the source is a T-Domain model and a set of cards, the built-in library is runtime codes to implement dynamic behaviors of cards, and the target is a training system SmartTrainer.

2.4 Ontology literacy and ontology-aware authoring environment

The separation between conceptual-level and lexical level is very important for designing an ontology-aware authoring tool, because it is key to fill the gap between authors and tools. The source of the gap comes from the difference between authors’ ontology-literacy and tool’s one. As we have discussed, we have remarkable human capability to read other’s intention between the lines. However, it could be double edged because we might misunderstand others because we are not good at building an exact conceptual model consistent with all the constraint over the model. We could say human has high ontology literacy at lexical-level but low at conceptual level. Oppositely, we could say authoring tools has high ontology literacy at conceptual level but low at lexical-level. How can we fill the gap and even maximize performance of both authors and tools?

![Figure 5 Snapshot of authoring process in an ontology-aware authoring tool](image)

To arrange the best collaboration between authors and tools, it is quite important to create an environment for author to describe the model easily and for tools to operate the model systematically. Needless to say, we can not ignore the benefit of graphical look-and-feel user interface to improve the easy-to-use of authoring tools. Figure 5 shows rough image of
ontology-aware authoring environment we have designed taking the literacy gap into consideration.

Figure 5 (a) shows a situation where the author specifies an training action using the action editor whose fields are arranged for the author easy to use. The statement filled each field are specified in lexical-level ontology as shown in figure 5 (b), which is an example of establish educational principle designed based on a instructional theory. As we have discussed, the lexical-level ontology is so organized in systematic manner that authors can easily reach the apt words to express their thoughts. Conceptual-level ontology specifies the detailed meaning of the statement as shown figure 5 (c). Note that the four statements in the editor’s fields imply the rather complex conceptual structure. In such an ontology-aware authoring environment, the author feels easy to describe his/her own thought and the tool can operate the corresponding conceptual model systematically.

2.5 Design Patterns

Axioms in an ontology give semantic constraints among concepts besides rigorous definition of concepts. They play a role of constraints which the models built have to satisfy. An ontology-aware authoring system thus can generate warning messages to the authors when a constraint is violated. The constraints could work as guidelines of the model-building process. This is the source of intelligence of such an authoring system. However, we have to pay an attention to the fact that not all the guidelines have to be satisfied by all the model-building processes. As described in 2.3.2, authoring still requires heuristics. That is, there can be soft heuristic guidelines which are sometimes better to satisfy but not a must, supported not by theory but by experiences and so forth. These kinds of suggestions are not appropriate for representation in the form of axioms. Axiomatic representation is appropriate only for theories.

In the object-oriented paradigm, design patterns[GAM94] are becoming popular recently. To avoid confusion between instructional "design" and "design" pattern, in this paper, we call the latter "heuristic design pattern".

Heuristic design patterns are patterns of class configuration which represent a chunk of information of useful suggestions about design decisions for configuring classes and objects. They are obtained by abstracting the rich experiences of object-oriented modeling and can provide suggestions useful during the course of model building. We can use heuristic design patterns of IIS modeling for the authoring systems on top of the ontology-guided and ID-theory-based modeling to generate helpful suggestions.

The use of heuristic design patterns in our plan is different from that in Object-oriented paradigm, given the declarative nature of an ontology. Heuristic design patterns in an ontology-aware authoring system are represented in terms of an ontology, and hence they are operational, while those in object-oriented paradigm are not operational because they are informal assuming human interpretation. All the terms appearing in an level 2 ontology are operational in the sense that computers can interpret the axioms associated with the terms used in the design patterns so that applicability of the patterns to the current situations can be checked by computers with appropriate suggestions. Application of design patterns to IESs are recently discussed by V. Devedzic[DEV99] and H.Shimizu, et al.[SHI99].

3. Towards theory-aware authoring systems

Adaptivity is the heart of intelligent systems. It comes from the declarative representation of what the system knows about the world it is in. In IES cases, the world consists of a learner and the system itself.

So, such a system behaves adaptively to the learner's understanding state. A learner model, which is a representation of system’s knowledge about the learner, serves as the source of intelligence. The system can investigate the learner model to adapt its behaviors to the
learner. In this sense, the model should be represented declaratively in order for the system to update and interpret it.

What about authoring systems? Do they have such a model or declarative representation of what they know? Unfortunately, the answer is No. This one of the major reasons why authoring systems are not intelligent.

Intelligence of authoring systems is not a specific issue to authoring system research but a general issue common to most AI-ED research, since it is deeply related to the knowledge about how to build intelligent educational systems.

As discussed above, the source of intelligence of the systems is declarative descriptions about what they know. They can dynamically inspect such declarative descriptions in order to adapt their behavior to situations where they are operating. We could say intelligent systems know what they know and what they are doing.

The next issue is what knowledge authoring systems should possess. Conventional authoring can be viewed as a kind of knowledge acquisition (KA) from teachers/instructors/trainers. This view gives us a few constructive suggestions to improve the performance of authoring systems. We could learn from the research of knowledge acquisition in knowledge-based systems community where KA from domain experts has been the bottleneck of expert system building. One of the major causes of the difficulty is due to the heavy dependence on heuristic knowledge which is hard to acquire, to manipulate and to justify. This is why model-based approaches are incorporated in modern knowledge-based systems which rely mainly on domain theory and model of the target system instead of heuristics.

In our case, analogously, "Model-based approach" is strongly needed to make IES building more scientific and principled. We need to depart from the "heuristics-based approach" and employ a new technology, that is, instructional design knowledge to enable the development process to follow instructional design decisions based on principled and theory-based knowledge.

Expertise is composed of heuristics and domain theories in general. Once it is extracted and is at hand, heuristic knowledge is easy to implement and resulting systems usually work well. Many knowledge-based systems have been built employing heuristic knowledge. One of the shortcomings of heuristic approach is that it is not principled and it ignores existing theories. Another way of building a knowledge-based system is to use domain theories which are in general objective and convincing. People can easily accept such systems that are based on theories.

Issue here, however, is that all the theories in many of the theory-based systems are built in the procedures. Not the system but the developer knows the theory. Such a system cannot be said to be "theory-aware" Authoring systems, which are kind of meta-system in the sense that it generates IESs, need to be "theory-aware" to be intelligent. Rich accumulation of instructional and learning theories should be used to make authoring systems knowledgeable. Declarative representation of those theories enables them to be called "theory-aware". We believe ID knowledge server and a heuristic design pattern server would realize an effective combination of theories and heuristics for intelligent authoring environments.

4. Concluding Remarks

We have discussed ontology-awareness of authoring tools. The most important role of task ontology is to lay the theoretical foundation for IES development process. It maintains continuity from author’s conceptual understanding of educational task to the computational semantics of IESs. It provides human friendly vocabulary for authors to describe the educational task. For the authoring tools, on the other hand, it specifies the computational
semantics of vocabulary and also provides a set of components represented in terms of both conceptual primitives and object-oriented code fragments.

The implication of "ontology-awareness" is deep. It is not only for AI-ED research for system building but for knowledge sharing among humans and computers. Most of the existing theories are in the form of informal representation assuming human interpretation. One of the most important merits introduced by ontological engineering is that it enables human to share the theories with computers. Furthermore, it can mediate people to find a minimum agreement.

We can hear some voice saying "It's OK that an ontology represents a shared conceptualization in a community. But, let me tell you that instructional design is the most dangerous domain to design an ontology because it is almost impossible to find a consensus." Our answer is: Yes, but, it is the very reason why we advocate ontological engineering of instructional design. Our claim includes an ontology helps people identify what they agree on and what they don't. An ontology is not the total knowledge of the target world but is a backbone/skeleton of the target world. Each person could come up with his/her own ontology developed by an ontology editor with sophisticated guidelines. It may be far from agreement, but it should be a good start to come up with richer agreement.

Reference: